Police API Client Documentation

Release 1.0.1

Rock Kitchen Harris

September 03, 2015

The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

CHAPTER 1

Contents

1.1 Police API Client (Python)

The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

1.1.1 Contents

Police API Client (Python)

The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Police API Client (Python) The Police API Client is an open-source client for the Police API. It was built to power the new Police.uk website.

View the README for installation instructions and quick-start examples.

Contents

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces``()`

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

`get_crime_category`(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **`id`** (*str*) – The ID of the crime category to get.
- **`date`** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

`get_crime`(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **`persistent_id`** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

`get_crimes_point`(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **`lat`** (*float or str*) – The latitude of the location.
- **`lng`** (*float or str*) – The longitude of the location.
- **`date`** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **`category`** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

`get_crimes_area`(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **`points`** (*list*) – A `list` of `(lat, lng)` tuples.
- **`date`** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **`category`** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

`get_crimes_location`(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
```

```
 u'url': u'http://www.leicester.gov.uk/'},
{u'description': None,
 u'title': u'Beaumont Leys LPU',
 u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
```

```
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.Officer`(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.

- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.SeniorOfficer`(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.Force`(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

`title`

Type str

The title of the event.

`type`

Type str

The type of the event.

`description`

Type str

A description of the event.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

`start_date`

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

`address`

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces``()`

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods``(`*force*`)`

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood``(`*force*, *neighbourhood*`)`

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces``()`

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods``(`*force*`)`

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood``(`*force*, *neighbourhood*`)`

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.SeniorOfficer`(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.Force`(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces``()`

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods``(`*force*`)`

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood``(`*force*, *neighbourhood*`)`

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces``()`

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods``(`*force*`)`

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood``(`*force*, *neighbourhood*`)`

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces`()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods`(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **`force`** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

`get_neighbourhood`(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location (*force*, *date=None*, *category=None*)
Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**`SeniorOfficer`**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

`force`

Type Force

The police force that this officer works for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
```

```
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.

- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
```

```
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`

- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories (*date=None*)
Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category (*id*, *date=None*)
Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class police_api.neighbourhoods.**Neighbourhood**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This crime category's slugified name.

`name`

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.PoliceAPI(**config)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.Event`(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

`name`

Type str

The officer's name.

`rank`

Type str

The officer's rank.

`bio`

Type str

The officer's biography.

`contact_details`

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`neighbourhood`

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

Configuration

The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

Forces

To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

Neighbourhoods

Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

Crime & Outcomes

The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
```

```
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

Reference

Police API

class `police_api.`**`PoliceAPI`**`(`***config*`)`

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **`base_url`** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **`user_agent`** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **`timeout`** – The timeout in seconds. Default: `30`
- **`username`** – The username to authenticate with. Default: `None`
- **`password`** – The password to authenticate with. Default: `None`

`get_forces``()`

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

`get_neighbourhoods``(`*force*`)`

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime (*persistent_id*)
Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point (*lat*, *lng*, *date=None*, *category=None*)
Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area (*points*, *date=None*, *category=None*)
Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location (*location_id*, *date=None*)
Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.

- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

Forces

Senior Officers

class `Force.`**SeniorOfficer**(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.`**`Force`**(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`preload`** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **`attrs`** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

`id`

Type str

The force's identifier (a slugified version of the name).

`name`

Type str

The full name of the force.

`description`

Type str

A short description of the force's role.

`url`

Type str

The force's website address.

`telephone`

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

`engagement_methods`

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
```

```
          u'title': u'twitter',
          u'type': u'twitter',
          u'url': u'http://www.twitter.com/leicspolice'},
         {u'description': None,
          u'title': u'youtube',
          u'type': u'youtube',
          u'url': u'http://www.youtube.com/leicspolice'},
         {u'description': None,
          u'title': u'rss',
          u'type': u'rss',
          u'url': u'http://www.leics.police.uk/feeds/news/'},
         {u'description': None,
          u'title': u'telephone',
          u'type': u'telephone',
          u'url': u''},
         {u'description': None,
          u'title': u'flickr',
          u'type': u'flickr',
          u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.Officer`(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.`**`Priority`**(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.`**`Neighbourhood`**(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.

- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This outcome category's slugified name.

name

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This location's unique ID.

name

Type str

The name of this location (e.g. `On or near Petrol Station`)

latitude

Type str

This location's latitude.

longitude

Type str

This location's longitude.

type

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

is_btp()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

crime

Type Crime

The crime that this outcome refers to.

category

Type OutcomeCategory

The category of this particular outcome.

date

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

persistent_id

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

month

Type str

The month that this crime was reported in (`%m-%d`).

category

Type CrimeCategory

The category of this crime.

location

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

1.1.2 Configuration

The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

1.1.3 Forces

To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

1.1.4 Neighbourhoods

Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

Officers

The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

Events

Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

Priorities

Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

1.1.5 Crime & Outcomes

The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

1.2 Reference

1.2.1 Police API

class `police_api.`**`PoliceAPI`**(***config*)

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI(user_agent='cops-and-robbers/9.9.9', timeout=60)
```

Parameters

- **base_url** – The base endpoint URL for the Police API. Default: `'http://data.police.uk/api/'`
- **user_agent** – The user agent string to use. Default: `'police-api-client-python/<version>'`
- **timeout** – The timeout in seconds. Default: `30`
- **username** – The username to authenticate with. Default: `None`
- **password** – The password to authenticate with. Default: `None`

get_forces()

Get a list of all police forces. Uses the forces API call.

Return type list

Returns A list of Force objects (one for each police force in England, Wales and Northern Ireland).

get_neighbourhoods(*force*)

Get a list of all neighbourhoods for a force. Uses the neighbourhoods API call.

Parameters **force** (*str or Force*) – The force to get neighbourhoods for (either by ID or Force object)

Return type list

Returns A `list` of Neighbourhood objects (one for each Neighbourhood Policing Team in the given force).

get_neighbourhood(*force*, *neighbourhood*)

Get a specific neighbourhood. Uses the neighbourhood API call.

Parameters

- **force** (*str or Force*) – The force within which the neighbourhood resides (either by ID or Force object)
- **neighbourhood** (*str*) – The ID of the neighbourhood to fetch.

Return type *Neighbourhood*

Returns The Neighbourhood object for the given force/ID.

locate_neighbourhood(*lat*, *lng*)

Find a neighbourhood by location. Uses the locate-neighbourhood API call.

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.

Return type Neighbourhood or None

Returns The Neighbourhood object representing the Neighbourhood Policing Team responsible for the given location.

get_dates()

Get a list of available dates. Uses the crimes-street-dates API call.

Return type list

Returns A `list` of `str` representing each monthly data set, in the format `YYYY-MM`, most recent first.

get_latest_date()

Get the latest available date. Uses the crimes-street-dates API call (not crime-last-updated, becuase the format differs).

Return type str

Returns The most recent data set's date, in the format `YYYY-MM`.

get_crime_categories(*date=None*)

Get a list of crime categories, valid for a particular date. Uses the crime-categories API call.

Return type list

Parameters **date** (*str or None*) – The date of the crime categories to get.

Returns A `list` of crime categories which are valid at the specified date (or at the latest date, if `None`).

get_crime_category(*id*, *date=None*)

Get a particular crime category by ID, valid at a particular date. Uses the crime-categories API call.

Return type *CrimeCategory*

Parameters

- **id** (*str*) – The ID of the crime category to get.
- **date** (*str or None*) – The date that the given crime category is valid for (the latest date is used if `None`).

Returns A crime category with the given ID which is valid for the specified date (or at the latest date, if `None`).

get_crime(*persistent_id*)

Get a particular crime by persistent ID. Uses the outcomes-for-crime API call.

Return type *Crime*

Parameters **persistent_id** (*str*) – The persistent ID of the crime to get.

Returns The `Crime` with the given persistent ID.

get_crimes_point(*lat*, *lng*, *date=None*, *category=None*)

Get crimes within a 1-mile radius of a location. Uses the crime-street API call.

Return type list

Parameters

- **lat** (*float or str*) – The latitude of the location.
- **lng** (*float or str*) – The longitude of the location.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within 1 mile of the specified location, in the given month (optionally filtered by category).

get_crimes_area(*points*, *date=None*, *category=None*)

Get crimes within a custom area. Uses the crime-street API call.

Return type list

Parameters

- **points** (*list*) – A `list` of `(lat, lng)` tuples.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported within the specified boundary, in the given month (optionally filtered by category).

get_crimes_location(*location_id*, *date=None*)

Get crimes at a particular snap-point location. Uses the crimes-at-location API call.

Return type list

Parameters

- **location_id** (*int*) – The ID of the location to get crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).

Returns A `list` of crimes which were snapped to the location with the specified ID in the given month.

get_crimes_no_location(*force*, *date=None*, *category=None*)

Get crimes with no location for a force. Uses the crimes-no-location API call.

Return type list

Parameters

- **force** (*str or Force*) – The force to get no-location crimes for.
- **date** (*str or None*) – The month in which the crimes were reported in the format `YYYY-MM` (the latest date is used if `None`).
- **category** (*str or CrimeCategory*) – The category of the crimes to filter by (either by ID or CrimeCategory object)

Returns A `list` of crimes which were reported in the given month, by the specified force, but which don't have a location.

1.2.2 Forces

Senior Officers

class `Force.SeniorOfficer`(*api*, *data={}*)

A senior police officer. Uses the senior-officers API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

force

Type Force

The police force that this officer works for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = force.senior_officers[0]
>>> pprint(officer.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

class `police_api.forces.Force`(*api*, *preload=False*, ***attrs*)

A police force in England, Wales or Northern Ireland. Uses the force API call.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> force.name
'Leicestershire'
```

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `id` is required. Any other attributes supplied will be set on the instance and not fetched from the API.

id

Type str

The force's identifier (a slugified version of the name).

name

Type str

The full name of the force.

description

Type str

A short description of the force's role.

url

Type str

The force's website address.

telephone

Type str

The force's main switchboard number. Usually set to `'101'` since the introduction of the national service.

engagement_methods

Type list

A `list` of `dict`, containing the keys `url`, `type`, `description`, and `title`.

```
>>> pprint(force.engagement_methods)
[{u'description': None,
  u'title': u'facebook',
  u'type': u'facebook',
  u'url': u'http://www.facebook.com/leicspolice'},
 {u'description': None,
  u'title': u'twitter',
  u'type': u'twitter',
  u'url': u'http://www.twitter.com/leicspolice'},
 {u'description': None,
  u'title': u'youtube',
  u'type': u'youtube',
  u'url': u'http://www.youtube.com/leicspolice'},
 {u'description': None,
  u'title': u'rss',
  u'type': u'rss',
  u'url': u'http://www.leics.police.uk/feeds/news/'},
 {u'description': None,
  u'title': u'telephone',
  u'type': u'telephone',
  u'url': u''},
 {u'description': None,
  u'title': u'flickr',
  u'type': u'flickr',
  u'url': u'http://www.flickr.com/photos/leicspolice-property'}]
```

neighbourhoods

Type list

A `list` of `Neighbourhood` objects (all the Neighbourhood Policing Teams in this force area).

senior_officers

Type list

A `list` of `Force.SeniorOfficer` objects.

1.2.3 Neighbourhoods

Events

class `Neighbourhood.`**`Event`**(*api*, *data={}*)

A neighbourhood event (e.g. a beat meating or surgery). Uses the neighbourhood-events API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that organised this event.

title

Type str

The title of the event.

type

Type str

The type of the event.

description

Type str

A description of the event.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

start_date

Type list

A `list` of `dict`, containing methods of contacting the event organisers.

```
>>> event = neighbourhood.events[0]
>>> pprint(event.contact_details)
{u'twitter': u'http://www.twitter.com/CCLeicsPolice'}
```

address

Type str

The location of the event.

start_date

Type datetime.datetime

The date and time that the event starts.

```
>>> event = neighbourhood.events[0]
>>> event.start_date
datetime.datetime(2014, 7, 14, 9, 30)
```

Officers

class `Neighbourhood.`**`Officer`**(*api*, *data={}*)

A police officer. Uses the neighbourhood-team API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that this officer has responsibility for.

name

Type str

The officer's name.

rank

Type str

The officer's rank.

bio

Type str

The officer's biography.

contact_details

Type list

A `list` of `dict`, containing methods of contacting the officer.

```
>>> officer = neighbourhood.officers[0]
>>> pprint(officer.contact_details)
{u'telephone': u'01788 853851',
 u'website': u'http://www.safer-neighbourhoods.co.uk/your-neighbourhood/rugby-district/rugby
```

Priorities

class `Neighbourhood.Priority`(*api*, *data={}*)

A neighbourhood priority (i.e. an issue raised by the community and a corresponding policing action to address this). Uses the neighbourhood-priorities API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

neighbourhood

Type Neighbourhood

The Neighbourhood Policing Team that owns this priority.

issue

Type str

The issue that was raised.

action

Type str

The action that was taken to address the issue.

issue_date

Type datetime.datetime

The date that the issue was raised.

action_date

Type datetime.datetime

The date that the action was implemented.

class `police_api.neighbourhoods.Neighbourhood`(*api*, *preload=False*, ***attrs*)

A Neighbourhood Policing Team. Uses the neighbourhood API call.

Parameters

- **api** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **preload** (*bool*) – If `True`, attributes are loaded from the API on instantiation rather than waiting for a property to be accessed.
- **attrs** – Only the `force` and `id` are required. Any other attributes supplied will be set on the instance and not fetched from the API.

```
>>> from police_api import PoliceAPI
>>> from police_api.forces import Force
>>> from police_api.neighbourhoods import Neighbourhood
>>> api = PoliceAPI()
>>> force = Force(api, id='leicestershire')
>>> neighbourhood = Neighbourhood(api, force=force, id='C04')
>>> neighbourhood.name
'City Centre Neighbourhood'
```

id

Type str

The neighbourhood's identifier (usually a code, but can contain spaces).

name

Type str

The name of the NPT.

description

Type str

A description of the NPT's area.

url_force

Type str

The URL for this NPT on the force's website

population

Type str

An estimate of the number of people living within the NPT boundary.

centre

Type dict

The approximate centre point of the neighbourhood.

```
>>> neighbourhood.centre
{"latitude": "52.6268", "longitude": "-1.12621"}
```

links

Type list

A `list` of links relevant to this force.

```
>>> pprint(neighbourhood.links)
[{u'description': None,
  u'title': u'Leicester City Council',
  u'url': u'http://www.leicester.gov.uk/'},
 {u'description': None,
  u'title': u'Beaumont Leys LPU',
  u'url': u'http://leicspolice.wordpress.com/category/lpu-blogs/beaumont-leys/'}]
```

locations

Type list

A `list` of police stations in this NPT.

```
>>> pprint(neighbourhood.locations)
[{u'address': u'2 Beaumont Way\n, Leicester',
  u'description': None,
  u'latitude': None,
  u'longitude': None,
  u'name': u'Beaumont Leys',
  u'postcode': u'LE4 1DS',
  u'type': u'station'}]
```

contact_details

Type dict

Ways that this NPT can be contacted.

```
>>> pprint(neighbourhood.contact_details)
{u'email': u'beaumont.lpu@leicestershire.pnn.police.uk',
 u'facebook': u'http://www.facebook.com/leicestercitypolice',
 u'telephone': u'101',
 u'twitter': u'http://www.twitter.com/LPAbbey'}
```

officers

Type list

A `list` of `Neighbourhood.Officer` objects.

events

Type list

A `list` of `Neighbourhood.Event` objects.

priorities

Type list

A `list` of `Neighbourhood.Priority` objects.

boundary

Type list

A `list` of `(lat, lng)` coordinates representing the perimeter of this neighbourhood's boundary.

```
>>> pprint(neighbourhood.boundary)
[(52.6235790036, -1.1433951806),
 (52.6235759765, -1.1432002292),
 ...
 (52.6241719477, -1.143313233),
 (52.6235790036, -1.1433951806)]
```

1.2.4 Crime

Crime Categories

class `police_api.crime.`**`CrimeCategory`**(*api*, *data={}*)

A crime category. Uses the crime-categories API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

id

Type str

This crime category's slugified name.

name

Type str

The name of this crime category.

Outcome Categories

class `police_api.crime.`**`OutcomeCategory`**(*api*, *data={}*)

An outcome category.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type str

This outcome category's slugified name.

`name`

Type str

The name of this outcome category.

Locations

class `police_api.crime.`**`Location`**(*api*, *data={}*)

An anonymised location, to which crimes are "snapped".

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This location's unique ID.

`name`

Type str

The name of this location (e.g. `On or near Petrol Station`)

`latitude`

Type str

This location's latitude.

`longitude`

Type str

This location's longitude.

`type`

Type str

This location's type (either `Force` or `BTP`, indicating whether the location belongs to a police force or a railway station).

`is_btp`()

Return type bool

Returns `True` if this location's type is `BTP`, and `False` otherwise.

Outcomes

class `Crime.`**`Outcome`**(*api*, *data={}*)

An outcome for an individual crime.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`crime`

Type Crime

The crime that this outcome refers to.

`category`

Type OutcomeCategory

The category of this particular outcome.

`date`

Type str

The month that this outcome was recorded in (`%m-%d`).

class `police_api.crime.`**`Crime`**(*api*, *data={}*)

An individual crime. Uses the outcomes-for-crime API call.

Parameters

- **`api`** (PoliceAPI) – The instance of `PoliceAPI` to use.
- **`data`** (*dict*) – The attributes that will be copied to this instance.

`id`

Type int

This crime's unique internal ID (not used elsewhere in the data or API).

`persistent_id`

Type str

This crime's persistent ID, which is referenced by the outcomes data and in the CSV files. Not guaranteed to be unique.

`month`

Type str

The month that this crime was reported in (`%m-%d`).

`category`

Type CrimeCategory

The category of this crime.

`location`

Type Location

The anonymised location that this crime occurred closest to.

context

Type str

Additional data about this crime provided by the reporting force.

outcome_status

Type Crime.Outcome

The latest outcome to have been recorded for this crime.

outcomes

Type list

A `list` of `Outcome` objects for this crime.

CHAPTER 2

Configuration

The API doesn't require any configuration or authentication, so all you need to do to get going is make a PoliceAPI instance:

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
```

Fore available methods and configuration parameters, see the Police API reference.

CHAPTER 3

Forces

To retrieve a list of police forces, use `get_forces()`:

```
>>> api.get_forces()
[<Force> Avon and Somerset Constabulary, ..., <Force> Wiltshire Police]
```

If you know the ID of a particular force, then you can use `get_force()`:

```
>>> force = api.get_force('leicestershire')
>>> force
<Force> Leicestershire Police
```

Fore available attributes and methods, see the Forces reference.

CHAPTER 4

Neighbourhoods

Forces are broken down into *Neighbourhood Policing Teams*:

```
>>> force.neighbourhoods
[<Neighbourhood> C02, <Neighbourhood> L03, ..., <Neighbourhood> L69]
```

If you know the ID of a particular neighbourhood, then you can use `get_neighbourhood()`:

```
>>> neighbourhood = api.get_neighbourhood('leicestershire', 'C02')
>>> neighbourhood
<Neighbourhood> C02
```

Or, if you already have a Force object:

```
>>> neighbourhood = force.get_neighbourhood('C02')
>>> neighbourhood
<Neighbourhood> C02
```

Fore available attributes and methods, see the Neighbourhoods reference.

4.1 Officers

The contact details for each officer in a particular neighbourhood are available:

```
>>> neighbourhood.officers
[<Neighbourhood.Officer> Michelle Zakoscielny, ..., <Neighbourhood.Officer> Richard Jones]
```

Fore available attributes and methods, see the Officers reference.

4.2 Events

Neighbourhood-level events (beat meetings, surgeries, etc.) are available:

```
>>> neighbourhood.events
[<Neighbourhood.Event> Stocking Farm beat surgery, ..., <Neighbourhood.Event> Stocking Farm beat surg
```

Fore available attributes and methods, see the Events reference.

4.3 Priorities

Policing teams set priorities to deal with in their neighbourhoods, which are represented by an *issue*, and an *action* to be taken:

```
>>> neighbourhood.priorities
[<Neighbourhood.Priority> <p>To address the issues of people begging next to cash machines in Market
```

Fore available attributes and methods, see the Priorities reference.

CHAPTER 5

Crime & Outcomes

The crime data is updated monthly, and each data set is represented by a date string, in the format `YYYY-MM`:

```
>>> api.get_dates()
[u'2014-03', u'2014-02', u'2014-01', ..., u'2010-12']
>>> api.get_latest_date()
u'2014-03'
```

To get crimes within a particular neighbourhood, call `get_crimes_area` with that neighbourhood's boundary:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary))
[<Crime> 30412621,
 <Crime> 30412622,
 <Crime> 30409577,
 <Crime> 30411516,
 ...
 <Crime> 30410475,
 <Crime> 30412775,
 <Crime> 30411518,
 <Crime> 30412182]
```

To fetch data for months other than the latest one, use a date string like the ones returned by `get_dates`:

```
>>> pprint(api.get_crimes_area(neighbourhood.boundary, date='2013-10'))
[<Crime> 27566767,
 <Crime> 27573059,
 <Crime> 27570299,
 <Crime> 27570923,
 ...
 <Crime> 27569847,
 <Crime> 27570896,
 <Crime> 27571396,
 <Crime> 27570916]
```

Crimes contain the date, category and location:

```
>>> crime = api.get_crime('ddf4c172d29569ab0cb667a346bcffad18f54a9bc3e0ae9694d2daf6738f068b')
>>> crime
<Crime> 20325597
>>> crime.month
u'2013-01'
>>> crime.category
<CrimeCategory> Shoplifting
>>> crime.location
<Location> 701166
```

```
>>> crime.location.name, crime.location.latitude, crime.location.longitude
(u'On or near Constance Close', u'51.737837', u'-2.235178')
```

Crimes have a list of outcomes, which represents the timeline of events since the crime was reported:

```
>>> pprint(crime.outcomes)
[<Crime.Outcome> Under investigation,
 <Crime.Outcome> Suspect charged,
 <Crime.Outcome> Awaiting court outcome,
 <Crime.Outcome> Offender imprisoned]
>>> crime.outcomes[-1].date
u'2013-01'
```

Crime objects representing Anti-Social Behaviour will not have outcomes:

```
>>> asb = api.get_crimes_area(neighbourhood.boundary, category='anti-social-behaviour')[0]
>>> asb.outcomes
[]
```

Fore available attributes and methods, see the Crime reference.

A

B

C

D

E

F

G

I

L

M

N

O

P

R

S

T

U