
Police API Client Documentation

Release 1.0.1

Rock Kitchen Harris

September 11, 2015

| | | |
|----------|-----------------------------|-----------|
| 1 | Reference | 3 |
| 1.1 | Police API | 3 |
| 1.2 | Forces | 6 |
| 1.3 | Neighbourhoods | 8 |
| 1.4 | Crime | 11 |
| 1.5 | Stop and search | 14 |
| 2 | Configuration | 17 |
| 3 | Forces | 19 |
| 4 | Neighbourhoods | 21 |
| 4.1 | Officers | 21 |
| 4.2 | Events | 21 |
| 4.3 | Priorities | 22 |
| 5 | Crime & Outcomes | 23 |
| | Python Module Index | 25 |

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 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: 'https://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 [forces.Force](#) objects (one for each police force represented in the API)

get_force(id, **attrs)

Get an individual forces. Uses the [force](#) API call.

Parameters **id** – The ID of the force to get information about.

Return type [forces.Force](#)

Returns The appropriate [forces.Force](#) object.

get_neighbourhoods(force)

Get a list of all neighbourhoods for a force. Uses the [neighbourhoods](#) API call.

Parameters **force** (str or [forces.Force](#)) – The force to get neighbourhoods for (either by ID or [forces.Force](#) object)

Return type list

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

get_neighbourhood (*force, id, **attrs*)

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 *forces.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](#), because 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 *Crime* objects 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 [crime.NoLocationCrime](#) objects which were reported in the given month, by the specified force, but which don't have a location.

1.2 Forces

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

A police force.

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

Parameters

- **api** ([PoliceAPI](#)) – The API instance 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.

```
>>> from pprint import pprint
>>> pprint(['{type}: {url}'.format(**method)
...         for method in force.engagement_methods])
['facebook: http://www.facebook.com/leicspolice',
 'twitter: http://www.twitter.com/leicspolice',
 'youtube: http://www.youtube.com/leicspolice',
 'rss: http://www.leics.police.uk/feeds/news/',
 'telephone: ',
 'flickr: 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.

class SeniorOfficer (*api*, *data*={})

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

Parameters

- **api** (*PoliceAPI*) – The API instance 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.

```
>>> from police_api import PoliceAPI
>>> force = PoliceAPI().get_force('leicestershire')
>>> officer = force.senior_officers[0]
>>> print(officer.contact_details['twitter'])
http://www.twitter.com/CCLeicsPolice
```

1.3 Neighbourhoods

class `police_api.neighbourhoods.Neighbourhood` (*args, **kwargs)
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
>>> api = PoliceAPI()
>>> force = api.get_force('leicestershire')
>>> neighbourhood = force.get_neighbourhood('C04')
>>> print(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.

```
>>> print(neighbourhood.centre['latitude'])
52.6268
>>> print(neighbourhood.centre['longitude'])
-1.12621
```

links**Type** list

A list of links relevant to this force.

```
>>> link = neighbourhood.links[0]
>>> print(link['title'])
Leicester City Council
>>> print(link['url'])
http://www.leicester.gov.uk/
```

locations**Type** list

A list of police stations in this NPT.

```
>>> print(neighbourhood.locations[0]['address'])
74 Belgrave Gate
, Leicester
```

contact_details**Type** dict

Ways that this NPT can be contacted.

```
>>> print(neighbourhood.contact_details['email'])
centralleicester.npa@leicestershire.pnn.police.uk
>>> print(neighbourhood.contact_details['twitter'])
http://www.twitter.com/leicesterpolice
```

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.

```
>>> neighbourhood.boundary[0]
(52.6235790036, -1.1433951806)
```

```
class 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.

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
>>> force = api.get_force('surrey')
>>> neighbourhood = force.get_neighbourhood('ELCO')
>>> officer = neighbourhood.officers[0]
```

neighbourhood

Type [Neighbourhood](#)

The Neighbourhood Policing Team that this officer is part of.

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.

```
>>> print(officer.contact_details['email'])
elmbridge@surrey.pnn.police.uk
>>> print(officer.contact_details['telephone'])
101
```

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

A neighbourhood event (e.g. a beat meeting 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.

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
>>> force = api.get_force('leicestershire')
>>> neighbourhood = force.get_neighbourhood('C04')
>>> event = neighbourhood.events[0]
```

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.

address
 Type str
 The location of the event.

start_date
 Type datetime.datetime
 The date and time that the event starts.

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.

1.4 Crime

class `police_api.crime.Crime` (*api*, *data*={})
 An individual crime. Uses the [outcomes-for-crime](#) API call.

Parameters

- **api** ([PoliceAPI](#)) – The API instance 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, in the order they occurred.

class Outcome (*api*, *data*={})

An outcome for an individual crime.

Parameters

- **api** (*PoliceAPI*) – The API instance 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.Location(*args, **kwargs)
```

An anonymised location, to which crimes are “snapped”. Information about how location anonymisation works is published on the [data.police.uk about page](#).

Parameters

- **api** ([PoliceAPI](#)) – The API instance 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 ‘BTP’ or ‘Force’, indicating whether the location contains crimes snapped from the British Transport Police or all other forces).

is_btp()

Return type bool

Returns True if this location’s type is ‘BTP’, and False otherwise.

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

A crime category. Uses the [crime-categories](#) API call.

Parameters

- **api** ([PoliceAPI](#)) – The API instance to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id

Type str

A slug representing this crime category.

name

Type str

The name of this crime category.

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

An outcome category.

Parameters

- **api** ([PoliceAPI](#)) – The API instance to use.
- **data** (*dict*) – The attributes that will be copied to this instance.

id**Type** str

A slug representing this outcome category.

name**Type** str

The name of this outcome category.

class `police_api.crime.NoLocationCrime` (*api*, *data*={})A crime with no location. Retrieved via the [crimes-no-location](#) API call.

1.5 Stop and search

class `police_api.stop_and_search.Stop` (*api*, *data*={})

A stop and search incident. Only a few of the attributes here are guaranteed to be provided by forces, so take care around any None values you may encounter.

```
>>> from police_api import PoliceAPI
>>> api = PoliceAPI()
>>> def sort_stops_by_date(unsorted_stops):
...     return sorted(unsorted_stops, key=lambda s: s.datetime)
>>> stops = sort_stops_by_date(
...     api.get_stops_force('metropolitan', '2015-07')
... )
```

age_range**Type** str

Human-readable string representing the age range of the person stopped.

```
>>> print(stops[0].age_range)
25-34
```

object_of_search**Type** str

The officer's justification for conducting the search.

outcome**Type** str

The outcome of the stop.

```
>>> print(stops[0].outcome)
Offender given drugs possession warning
```

outcome_linked_to_object_of_search**Type** bool

Whether the outcome of the stop was related to the reason the stop was conducted.

legislation**Type** str

The legislation allowing this particular stop.

```
>>> print(stops[0].legislation)
Misuse of Drugs Act 1971 (section 23)
```

type**Type** str

What type of search this was (person, vehicle, etc.).

```
>>> print(stops[0].type)
Person search
```

involved_person**Type** bool

Whether or not a person was searched in this stop.

```
>>> stops[0].involved_person
True
>>> vehicle_stop = [
...     s for s in stops if s.type == 'Vehicle search'
... ][0]
>>> vehicle_stop.involved_person
False
```

operation**Type** bool

Whether this stop was part of a policing operation.

operation_name**Type** str

The name of the policing operation this stop was part of, if applicable.

self_defined_ethnicity**Type** str

The ethnicity of the person stopped, as reported by the person stopped.

```
>>> print(stops[0].self_defined_ethnicity)
Black or Black British - Any other Black ethnic background (B9)
```

officer_defined_ethnicity**Type** str

The ethnicity of the person stopped, as reported by the officer who conducted the stop.

```
>>> print(stops[0].officer_defined_ethnicity)
Black
```

gender**Type** str

The gender of the person stopped. It is not clear if this is as reported by the officer or the person stopped.

```
>>> print(stops[0].gender)
Male
```

datetime

Type datetime

When the stop was conducted. Note that if a force appears to only conduct stops at midnight, that probably means they don't record the time of stops.

```
>>> print(stops[0].datetime.isoformat())
2015-07-01T00:05:00
```

location

Type Location

The approximate location of the stop.

removal_of_more_than_outer_clothing

Type bool

Whether significant clothing was removed in order to carry out the search.

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()
```

For available methods and configuration parameters, see the [PoliceAPI](#) reference.

Forces

To retrieve a list of police forces, use *PoliceAPI.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 *PoliceAPI.get_force()*:

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

For available attributes and methods, see the *forces.Force* 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 `PoliceAPI.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
```

For available attributes and methods, see the `neighbourhoods.Neighbourhood` 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]
```

For available attributes and methods, see the `neighbourhoods.Neighbourhood.Officer` 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 surgery]
```

For available attributes and methods, see the `neighbourhoods.Neighbourhood.Event` 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
```

For available attributes and methods, see the [neighbourhoods.Neighbourhood.Priority](#) 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 `PoliceAPI.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 `PoliceAPI.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
[]
```

For available attributes and methods, see the [crime.Crime](#) reference.

p

- `police_api`, [3](#)
- `police_api.crime`, [11](#)
- `police_api.forces`, [6](#)
- `police_api.neighbourhoods`, [8](#)
- `police_api.stop_and_search`, [14](#)

A

action (police_api.neighbourhoods.Neighbourhood.Priority attribute), 11
action_date (police_api.neighbourhoods.Neighbourhood.Priority attribute), 11
address (police_api.neighbourhoods.Neighbourhood.Event attribute), 11
age_range (police_api.stop_and_search.Stop attribute), 14

B

bio (police_api.forces.Force.SeniorOfficer attribute), 7
bio (police_api.neighbourhoods.Neighbourhood.Officer attribute), 10
boundary (police_api.neighbourhoods.Neighbourhood attribute), 9

C

category (police_api.crime.Crime attribute), 12
category (police_api.crime.Crime.Outcome attribute), 12
centre (police_api.neighbourhoods.Neighbourhood attribute), 8
contact_details (police_api.forces.Force.SeniorOfficer attribute), 7
contact_details (police_api.neighbourhoods.Neighbourhood attribute), 9
contact_details (police_api.neighbourhoods.Neighbourhood.Officer attribute), 10
context (police_api.crime.Crime attribute), 12
Crime (class in police_api.crime), 11
crime (police_api.crime.Crime.Outcome attribute), 12
Crime.Outcome (class in police_api.crime), 12
CrimeCategory (class in police_api.crime), 13

D

date (police_api.crime.Crime.Outcome attribute), 12
datetime (police_api.stop_and_search.Stop attribute), 16
description (police_api.forces.Force attribute), 6
description (police_api.neighbourhoods.Neighbourhood attribute), 8

description (police_api.neighbourhoods.Neighbourhood.Event attribute), 11

E

engagement_methods (police_api.forces.Force attribute), 7
events (police_api.neighbourhoods.Neighbourhood attribute), 9

F

Force (class in police_api.forces), 6
force (police_api.forces.Force.SeniorOfficer attribute), 7
Force.SeniorOfficer (class in police_api.forces), 7

G

gender (police_api.stop_and_search.Stop attribute), 15
get_crime() (police_api.PoliceAPI method), 5
get_crime_categories() (police_api.PoliceAPI method), 4
get_crime_category() (police_api.PoliceAPI method), 4
get_crimes_area() (police_api.PoliceAPI method), 5
get_crimes_location() (police_api.PoliceAPI method), 5
get_crimes_no_location() (police_api.PoliceAPI method), 5
get_crimes_point() (police_api.PoliceAPI method), 5
get_dates() (police_api.PoliceAPI method), 4
get_force() (police_api.PoliceAPI method), 3
get_forces() (police_api.PoliceAPI method), 3
get_latest_date() (police_api.PoliceAPI method), 4
get_neighbourhood() (police_api.PoliceAPI method), 4
get_neighbourhoods() (police_api.PoliceAPI method), 3

I

id (police_api.crime.Crime attribute), 11
id (police_api.crime.CrimeCategory attribute), 13
id (police_api.crime.Location attribute), 13
id (police_api.crime.OutcomeCategory attribute), 14
id (police_api.forces.Force attribute), 6
id (police_api.neighbourhoods.Neighbourhood attribute), 8

involved_person (police_api.stop_and_search.Stop attribute), 15
is_btp() (police_api.crime.Location method), 13
issue (police_api.neighbourhoods.Neighbourhood.Priority attribute), 11
issue_date (police_api.neighbourhoods.Neighbourhood.Priority attribute), 11

L

latitude (police_api.crime.Location attribute), 13
legislation (police_api.stop_and_search.Stop attribute), 14
links (police_api.neighbourhoods.Neighbourhood attribute), 9
locate_neighbourhood() (police_api.PoliceAPI method), 4
Location (class in police_api.crime), 12
location (police_api.crime.Crime attribute), 12
location (police_api.stop_and_search.Stop attribute), 16
locations (police_api.neighbourhoods.Neighbourhood attribute), 9
longitude (police_api.crime.Location attribute), 13

M

month (police_api.crime.Crime attribute), 12

N

name (police_api.crime.CrimeCategory attribute), 13
name (police_api.crime.Location attribute), 13
name (police_api.crime.OutcomeCategory attribute), 14
name (police_api.forces.Force attribute), 6
name (police_api.forces.Force.SeniorOfficer attribute), 7
name (police_api.neighbourhoods.Neighbourhood attribute), 8
name (police_api.neighbourhoods.Neighbourhood.Officer attribute), 10
Neighbourhood (class in police_api.neighbourhoods), 8
neighbourhood (police_api.neighbourhoods.Neighbourhood.Event attribute), 10
neighbourhood (police_api.neighbourhoods.Neighbourhood.Officer attribute), 10
neighbourhood (police_api.neighbourhoods.Neighbourhood.Priority attribute), 11
Neighbourhood.Event (class in police_api.neighbourhoods), 10
Neighbourhood.Officer (class in police_api.neighbourhoods), 9
Neighbourhood.Priority (class in police_api.neighbourhoods), 11
neighbourhoods (police_api.forces.Force attribute), 7
NoLocationCrime (class in police_api.crime), 14

O

object_of_search (police_api.stop_and_search.Stop attribute), 14
officer_defined_ethnicity (police_api.stop_and_search.Stop attribute), 15
officers (police_api.neighbourhoods.Neighbourhood attribute), 9
operation (police_api.stop_and_search.Stop attribute), 15
operation_name (police_api.stop_and_search.Stop attribute), 15
outcome (police_api.stop_and_search.Stop attribute), 14
outcome_linked_to_object_of_search (police_api.stop_and_search.Stop attribute), 14
outcome_status (police_api.crime.Crime attribute), 12
OutcomeCategory (class in police_api.crime), 13
outcomes (police_api.crime.Crime attribute), 12

P

persistent_id (police_api.crime.Crime attribute), 12
police_api (module), 1, 3
police_api.crime (module), 11
police_api.forces (module), 6
police_api.neighbourhoods (module), 8
police_api.stop_and_search (module), 14
PoliceAPI (class in police_api), 3
population (police_api.neighbourhoods.Neighbourhood attribute), 8
priorities (police_api.neighbourhoods.Neighbourhood attribute), 9

R

rank (police_api.forces.Force.SeniorOfficer attribute), 7
rank (police_api.neighbourhoods.Neighbourhood.Officer attribute), 10
removal_of_more_than_outer_clothing (police_api.stop_and_search.Stop attribute), 16

S

senior_officers (police_api.stop_and_search.Stop attribute), 15
senior_officers (police_api.forces.Force attribute), 7
start_date (police_api.neighbourhoods.Neighbourhood.Event attribute), 11
Stop (class in police_api.stop_and_search), 14

T

telephone (police_api.forces.Force attribute), 6
title (police_api.neighbourhoods.Neighbourhood.Event attribute), 10
type (police_api.crime.Location attribute), 13

type (police_api.neighbourhoods.Neighbourhood.Event
attribute), [10](#)

type (police_api.stop_and_search.Stop attribute), [15](#)

U

url (police_api.forces.Force attribute), [6](#)

url_force (police_api.neighbourhoods.Neighbourhood at-
tribute), [8](#)