Newsapi Documentation

Release 0.1.0

Ujjwal Gupta

Mar 28, 2017

Contents

1 newsapi 3
1.1 Installation . . . 3
1.2 Dependencies . . . 3
1.3 API . . . 3
1.4 Usage . . . 4
1.5 Usage . . . 6
1.6 Errors and Exceptions . . . 8

2 Indices and tables 9

Contents:

CHAPTER 1

newsapi

Python wrapper around `A JSON API for live news and blog headlines` (a.k.a. `News Api`): https://newsapi.org/

NOTE: This library and its author are not endorsed by or affiliated with NewsApi.org.

Installation

Using `pip`:

```
pip install newsapi
```

Dependencies

- requests

API

`newsapi` offers two classes `Articles` and `Sources` for the functionality of two endpoints:- *https://newsapi.org/v1/articles* and *https://newsapi.org/v1/sources* offered by `News Api` respectively.

Articles

News API Param	newsapi Param	Value	Description
`source`	`source`	string **(required)**	The identifer for the news source or blog you want headlines from
`apiKey`	`API_KEY`	string **(required)**	Your API key.
`sortBy`	`sort_by`	string **(optional)**	Specify which type of list you want. The possible options are **top**, **latest** and **popular**. **Note**: not all options are available for **all** sources. **Default**: **top**.

Methods

All methods of `Articles` are accessible via:

```
from newsapi.articles import articles

a = Articles(API_KEY="Your-api-key")

# a.<method>
```

Method	Parameters	Description	Returns
`get()`	source (required), sort_by (optional), attributes_format (optional Default:True)	Generic request to NewsApi (with source as required parameter, sort_by as optional).	`AttrDict`
`get_by_top()`	source (required)	Requests a list of the source's headlines sorted in the order they appear on its homepage.	`AttrDict`
`get_by_latest()`	source (required).	Requests a list of the source's headlines sorted in chronological order, newest first.	`AttrDict`
`get_by_popular()`	source (required).	Requests a list of the source's current most popular or currently trending headlines.	`AttrDict`

NOTE: By default all `newsapi` responses are formatted as `JSON`, `AttrDict` is just a wrapper around Dictionary which enables to do content.status as well as content['status'], but can't do it in nested dicts.

Usage

General Import

```
from newsapi.articles import Articles

a = Articles(API_KEY="Your-api-key")
```

a.get()

```
# get all the articles from the new web and sorted by top (default).
a.get(source="the-new-web")
```

a.get_by_popular()

```
# get all the articles from the new web and sorted by popular (front page).
a.get_by_popular(source="the-new-web")
```

apply the same logic for **get_by_top**() *and* **get_by_latest**().

Sources

News API Param	news-api Param	Value	Description
`category`	`category`	string **(optional)**	The category you would like to get sources for. **Possible options**: *business*, *entertainment*, *gaming*, *general*, *music*, *science-and-nature*, *sport*, *technology*. **Default**: **empty** *(all sources returned)*
`language`	`language`	string **(optional)**	The 2-letter ISO-639-1 code of the language you would like to get sources for. **Possible options**: *en*, *de*, *fr*. **Default**: **empty** *(all sources returned)*.
`country`	`country`	string **(optional)**	The 2-letter ISO 3166-1 code of the country you would like to get sources for. **Possible options**: *au*, *de*, *gb*, *in*, *it*, *us*. **Default**: **empty** *(all sources returned)*.

Methods

All methods are accessible via:

```
from newsapi.sources import Sources

s = Sources(API_KEY="Your-api-key")

# s.<method>
```

Method	Parameters	Description	Returns
`get()`	category (optional), language (optional), country (optional), attributes_format (optional Default:True).	Generic request to NewsApi to get sources as needed with optional params. (default : empty returns all sources	`AttrDict`
`all()`	No parameters needed.	wrapper around get() to get all sources unfiltered.	`AttrDict`
`get_by_category()`	category (required).	The category you would like to get sources for.	`AttrDict`
`get_by_language()`	language (required).	The 2-letter ISO-639-1 code of the language you would like to get sources for.	`AttrDict`
`get_by_country()`	country (required).	The 2-letter ISO 3166-1 code of the country you would like to get sources for.	`AttrDict`
`information()`	No parameters needed.	Sets up everything by sending an unfiltered request and then sorting it.	`Self`
`all_sorted_information()`	No parameters needed.	gives back all the sources.	`Array`
`all_categories()`	detailed (optional, Default: False, gives all the information of sources group by categories).	Gets all the categories available by newsapi and grouped with info if detailed set to true.	`dict_keys/Dict`
`all_languages()`	detailed (optional, Default: False, gives all the information of sources group by languages).	Gets all the languages available by newsapi and grouped with info if detailed set to true.	`dict_keys/Dict`
`all_countries()`	detailed (optional, Default: False, gives all the information of sources group by countries).	Gets all the countries available by newsapi and grouped with info if detailed set to true.	`dict_keys/Dict`
`all_base_information()`	No parameters needed.	gives back all the name, id pairs of the available sources offered by newsapi.	`Dict`
`all_ids()`	detailed (optional, Default: False, gives name, id pair of all the sources).	gives back all the ids of the available sources offered by newsapi.	`dict_values/Dict`
`all_names()`	detailed (optional, Default: False, gives name, url pair of all the sources).	gives back all the names of the available sources offered by newsapi.	`dict_keys/Dict`
`all_urls()`	detailed (optional, Default: False, gives name, url pair of all the sources).	gives back all the urls of the available sources offered by newsapi.	`dict_values/Dict`
`search()`	name (required, the name of the source you wanna search for).	gives back all the matches from the given name of the source to avaiable ones by newsapi with all the info.	`Array`

NOTE: By default all `newsapi` responses are formatted as `JSON`, `AttrDict` is just a wrapper around Dictionary which enables to do content.status as well as content['status'], but can't do it in nested dicts.

Usage

General Import

```
from newsapi.sources import Sources

s = Sources(API_KEY="Your-api-key")
```

s.get()

```
# get sources with category technology and language as en while originated from␣
↪country uk
s.get(category='technology', language='en', country='uk')
```

s.all()

```
# get all sources offered by newsapi
s.all()
```

s.get_by_category()

```
# get all sources offered by newsapi with category as general
s.get_by_category("general")
```

same logic can be applied to **get_by_language()** *and* **get_by_country()**

s.information()

Note : you need to invoke **information()** method only once and after then you can use any methods given below. chaining them or not is all upto each individual's preference.

```
# sets up everything and sorts the raw data.
s.information()

#then you can chain functions, so instead of above command use this.
#gets all the categories offered by newsapi.
s.information().all_categories()

#or just call it standalone like so.
s.information()
s.all_categories()

#using detailed parameter results in categories group with sources info like
s.information().all_categories(detailed=True)
#results in:
# ['general' : [{'id': "the-new-web", 'name': "The New Web"}, ...], 'sports': [{'id':
↪"bbc-sports", 'name': "The BBC Sports"},...], ...]
```

same logic can be applied for **all_languages()** and **all_countries()**, after invoking **information()** as shown above.

s.all_base_information()

```
# get all sources in the name, url pair dict format offered by newsapi
s.information().all_base_information()
```

same logic can be applied for **all_sorted_information()**.

s.all_ids()

```
# get all sources ids offered by newsapi
s.information().all_ids()
```

same goes for **all_names()** and **all_urls()** after invoking **information()** as shown above.

s.search()

```
# search by string
s.search('bbc')
#results with array containing all the sources which has 'bbc' string present in it␣
↪with all its info.
```

Errors and Exceptions

Under the hood, `newsapi` uses the requests library. For a listing of explicit exceptions raised by `requests`, see Requests: Errors and Exceptions.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search