
wikimapia_api Documentation

Release 0.1.0.dev1

Alexey Ovchinnikov

September 09, 2016

1	Getting Started with wikimapia_api	3
1.1	Installation	3
1.2	Configuring	3
1.3	Using	4
2	Indices and tables	7

Wikimapia is an open-content collaborative map project. This library is a Python (2.7, 3.3 and 3.4) implementation of wikimapia [api](#).

Getting Started with wikimapia_api

1.1 Installation

wikimapia_api is available on PyPi and the preferred method of install is using pip. Library is in prerelease stage, so use `--pre` option:

```
$ pip install --pre wikimapia_api
```

1.2 Configuring

wikimapia_api have several configuration options. You can set these options for entire library and then override any of them in the specific API calls. To configure library, use `API.config` as follows:

```
from wikimapia_api import API, Config

# You can assign dict with config options
API.config = {'key': 'YOUR_API_KEY', 'delay': 5000}

# Or set options separately
API.config.language = 'en'
API.config.compression = False

# Or create new config instance
API.config = Config(delay=2000, language='de')

# Also you can read config values
print(API.config.key)
print(API.config.url)
```

Below are all available options:

option type	default	value description	
key	string	'example'	Your Wikimapia API key
url	string	' http://api.wikimapia.org '	API url
language	string	'en'	Response language
delay	number	3000	Delay between requests in milliseconds
compression	boolean	True	Use compression (gzip)

1.3 Using

1.3.1 Common options

If result contains geographical coordinates they will be in EPSG:4326 format - spherical longitude and latitude coordinates. To get mercator coordinates (projected to flat map space) specify `mercator=True` or `options='mercator'` in function call.

Many functions returns a lot of data. To prevent unwanted traffic you can specify necessary data blocks with `data_blocks='main, geometry'` or `data_blocks=['main', 'geometry']`. For full list of available data blocks for each function refer to [api](#) documentation.

1.3.2 Languages

To get all available languages:

```
languages = API.languages
```

This will make *Api.Getlanguages* request to API.

1.3.3 Categories

To get a list of all categories or categories that matches specific name use:

```
categories = API.categories.all()
search = API.categories.all(name='school')
for x in search:
    print(x)
# to override config options
search = API.categories.all(name='school', language='de', delay=1000)
```

To get specific category by id:

```
category = API.categories[203]
# or to override options
category = API.categories.get(203, delay=1000)
```

1.3.4 Streets

Wikimapia API provide only one function *Street.Getbyid* to retrieve street info. So to get street by id:

```
street = API.streets[50]
# or to specify options
street = API.streets.get(203, language='ru', mercator=True)
```

1.3.5 Places

To get places inside bounding box:

```
# specify lon_min, lat_min, lon_max, lat_max
places = API.places.inside(37.54, 55.72, 37.65, 55.77, category=203)
print(len(places))
```


To get places inside specific tile:

```
# specify x, y and z coordinates of tile
places = API.places.in_tile(4953, 2567, 13, category=203)
print(len(places))
```

To get places nearest to specific location:

```
# specify lon and lat of location
places = API.places.nearest(37.54, 55.72, category=203)
print(places[0])
```

To search places near specific location:

```
# specify text query, lon and lat of location
places = API.places.search('school 779', 37.54, 55.72, category=203)
print(places[0])
```

To get specific place by id:

```
place = API.place[496457]
# or to specify options
place = API.places.get(496457, data_blocks=['main', 'photos'])
```

Indices and tables

- `genindex`
- `modindex`
- `search`