
PyMapia Documentation

Release 0.1.1

Rob Derksen

Nov 10, 2017

Contents

1	Installing PyMapia	3
2	PyMapia	5
3	Indices and tables	11
	Python Module Index	13

Contents:

CHAPTER 1

Installing PyMapia

`pip install pymapia` or `run python setup.py install` from the top directory


```
class pymapia.PyMapia(api_key)
```

```
get_place_by_id(id_, language=None, data_blocks=None, options=None)
```

Returns information about place. Place information is organized in data-blocks:

- **main:** main information about place: url, title, description, categories, if place is a building, if it's a region. Also if it is deleted.
- **geometry:** place geometry on map: polygon or rectangle
- **edit:** user_id and name of last editor and timestamp. If the place is in deletion state this info will be in the edit block also
- **location:** place location: lat/lon coordinates, north/south/east/west coordinates, zoom level, country, state, city id and name, WikiMapia Cityguide domain name, street id and name
- **attached:** places attached to selected one or parent place of selected one, only basic info: url, title, categories. Also if child place is deleted.
- **photos:** photos of current place: urls to thumb, big and fullsize photo, id, size, author id and name, date of photo uploading, last editor of this photo, photo status (deleted/active)
- **comments:** place comments: number, language of comment, author id, his ip and name, comment text, positive and negative votes, moderator id, name, and date of deletion if the comment was removed
- **translate:** languages available for selected place
- **similar_places, nearest_places, nearest_comments, nearest_streets, nearest_hotels:** extra information about objects around selected one

Parameters

- **id** (*int*) – The identifier of the object you want to get information about.
- **language** (*str*) – The specified language in ISO 639-1 format

- **data_blocks** (*list*) – A list of strings specifying which blocks of data you want to return: main, geometry, edit, location, attached, photos, comments, translate, similar_places, nearest_places, nearest_comments, nearest_streets, nearest_hotels
- **options** (*list*) –

A list of options:

- mercator - all coordinates will be in the Mercator format instead of the default format.

get_place_by_area (*bbox=None, tile_coordinates=None, language=None, data_blocks=None, options=None, count=5, page=1, category=None, category_or=None*)

Returns all places in the given boundary box optionally filtered by category parameter. Only basic information is available: id, title, url, location and polygon of each place. Location and polygon fields can be turned off with data-block parameter.

Parameters

- **bbox** (*dict*) – Bounding box as dict with the following keys: lon_min, lon_max, lat_min, lat_max
- **tile_coordinates** (*dict*) – Tile coordinates as dict with the following keys: x, y, z
- **language** (*str*) – The specified language in ISO 639-1 format
- **data_blocks** (*list*) – A list of strings specifying which blocks of data you want to return: main, geometry, edit, location, attached, photos, comments, translate, similar_places, nearest_places, nearest_comments, nearest_streets, nearest_hotels
- **options** (*list*) –

A list of options:

- mercator - all coordinates will be in the Mercator format instead of the default format.
- **count** (*int*) – This is a variable that determines the number of results per page. 5 is default (5 min, 100 max).
- **page** (*int*) – The page number. 1 is default.
- **category** (*list*) – The WikiMapia category code as a list with category ids or text queries in UTF-8: School, Church, etc. Several comma-separated categories may be stated with “AND” logic. It means that only objects which have all listed categories together would be returned. If you need OR logic use `category_or` parameter.
- **category_or** (*list*) – The WikiMapia category code as a list with category ids or text queries in UTF-8: School, Church, etc. Several comma-separated categories may be stated with “OR” logic. All objects which have any category from this list would be returned. At first all places with any of `category_or` categories are selected and then only those which have all `category` categories are left.

get_nearest_place (*lat, lon, language=None, data_blocks=None, options=None, page=1, count=5, category=None*)

Returns search results of objects, closest to the selected point. Optionally filtered by category parameter. Only basic information is available: id, title, url, location and polygon of each place and distance from selected coordinates to it. Location and polygon fields can be turned off with data-block parameter.

Parameters

- **lat** (*float*) – Coordinates of the “search point” as a float; lat means latitude (ϕ).
- **lon** (*float*) – Coordinates of the “search point” as a float; lon means longitude (λ).

- **language** (*str*) – The specified language in ISO 639-1 format
- **data_blocks** (*list*) – A list of strings specifying which blocks of data you want to return: main, geometry, edit, location, attached, photos, comments, translate, similar_places, nearest_places, nearest_comments, nearest_streets, nearest_hotels
- **options** (*list*) –

A list of options:

- mercator - all coordinates will be in the Mercator format instead of the default format.
- **page** (*int*) – The page number. 1 is default.
- **count** (*int*) – This is a variable that determines the number of results per page. 5 is default (5 min, 100 max).
- **category** (*list*) – The WikiMapia category code as a list with category ids or text queries in UTF-8: School, Church, etc. Several categories may be stated with “AND” logic. It means that only objects which have all listed categories together would be returned.

search_place (*query, lat, lon, language=None, data_blocks=None, options=None, page=1, count=5, category=None, category_or=None, distance=None*)

Returns search results of a given query, optionally filtered by category. Only basic information is available: id, title, url, location and polygon of each place and distance from selected coordinates to it. The function is very close to `get_nearest_place`.

Parameters

- **query** (*str*) – The query to search in WikiMapia (UTF-8).
- **lat** (*float*) – Coordinates of the “search point” as a float; lat means latitude (ϕ).
- **lon** (*float*) – Coordinates of the “search point” as a float; lon means longitude (λ).
- **language** (*str*) – The specified language in ISO 639-1 format
- **data_blocks** (*list*) – A list of strings specifying which blocks of data you want to return: main, geometry, edit, location, attached, photos, comments, translate, similar_places, nearest_places, nearest_comments, nearest_streets, nearest_hotels
- **options** (*list*) –

A list of options:

- mercator - all coordinates will be in the Mercator format instead of the default format.
- **count** (*int*) – This is a variable that determines the number of results per page. 5 is default (5 min, 100 max).
- **page** (*int*) – The page number. 1 is default.
- **category** (*list*) – The WikiMapia category code as a list with category ids or text queries in UTF-8: School, Church, etc. Several comma-separated categories may be stated with “AND” logic. It means that only objects which have all listed categories together would be returned. If you need OR logic use the `category_or` parameter.
- **category_or** (*list*) – The WikiMapia category code as a list with category ids or text queries in UTF-8: School, Church, etc. Several comma-separated categories may be stated with “OR” logic. All objects which have any category from this list would be returned. At first all places with any of `category_or` categories are selected and then only those which have all `category` categories are left.

- **distance** (*float*) – Search only places which are not farther from the requested point than distance. Float value in meters.

update_place (*polygons*, *language*='en', *title*=None, *description*=None, *wikipedia_url*=None, *is_building*=False, *tags_ids*=None, *street_id*=None, *street_name*=None, *building_number*=None, *revision_comment*=None)

Creates a new place or updates an existing one. This function is still in a stage of development now. The function works with WikiMapia Guest's privileges only: it is possible to create new but not to edit existing places.

Parameters

- **polygons** (*list*) – List of tuples with the geo-points: [(lat1, lon1), (lat2, lon2), ..., (lat_n, lon_n)]
- **language** (*str*) – The specified language in ISO 639-1 format. Defaults to 'en'
- **title** (*str*) – The place title.
- **description** (*str*) – Place description.
- **wikipedia_url** (*str*) – A link to wikipedia article about the place.
- **is_building** (*bool*) – Set the parameter to True if the place is a building (allow to attach places to this one). Defaults to False
- **tags_ids** (*list*) – The WikiMapia category code as a list with category ids
- **street_id** (*int*) – Street ID to set into place address field. Set this parameter to 0 to remove the place address or to create a new street (with the *street_name* parameter).
- **street_name** (*str*) – Create a new street with this name and set it as place address. Works if *street_id* is 0.
- **building_number** (*str*) – Building number, the part of place address. Works if *street_id* is 0.
- **revision_comment** (*str*) – A brief description of the committed changes.

get_street_by_id (*id*_, *language*=None, *data_blocks*=None, *options*=None)

Returns information about street. Street information is organized in data-blocks:

- **main**: main information about street: url, title, description. Also if it is deleted.
- **edit**: *user_id* and name of last editor and timestamp. If the street is in deletion state this info will be in the edit block also
- **location**: place location: lat/lon coordinates, north/south/east/west coordinates, zoom level, country, state, city id and name, WikiMapia Cityguide domain name
- **related**: places related to the selected street, only basic info: url, title, categories.
- **photos**: photos of current street: urls to thumb, big and fullsize photo, id, size, author id and name, date of photo uploading, last editor of this photo, photo status (deleted/active)
- **comments**: street comments: number, language of comment, author id, his ip and name, comment text, positive and negative votes, moderator id, name, and date of deletion if the comment was removed
- **translate**: languages available for selected street
- **similar_places**, **nearest_places**, **nearest_comments**, **nearest_streets**, **nearest_hotels**: extra information about objects around selected street

Parameters

- **id** (*int*) – The identifier of the object you want to get information about.
- **language** (*str*) – The specified language in ISO 639-1 format
- **data_blocks** (*list*) – A list of strings specifying which blocks of data you want to return: main, geometry, edit, location, attached, photos, comments, translate, similar_places, nearest_places, nearest_comments, nearest_streets, nearest_hotels
- **options** (*list*) –

A list of options:

- mercator - all coordinates will be in the Mercator format instead of the default format.

get_category_by_id (*id_*, *language*)

Returns all information about selected category.

Parameters

- **id** (*int*) – The identifier of the category you want to get information about.
- **language** (*str*) – This is specified language in ISO 639-1 format.

get_all_categories (*language=None*, *name=None*, *page=1*, *count=50*)

Returns WikiMapia approved categories list. Optionally searches categories by name or a part of name.

Parameters

- **language** (*str*) – The specified language in ISO 639-1 format.
- **name** (*str*) – WikiMapia category name (or a part of it) in UTF-8: name=School, name=Church etc. or a part of it for searching
- **page** (*int*) – The page number. 1 is default.
- **count** (*int*) – This is a variable that determines the number of results per page. 50 is default (5 min, 100 max).

CHAPTER 3

Indices and tables

- `genindex`
- `search`

p

`pymapia`, 5

G

`get_all_categories()` (pymapia.PyMapia method), 9
`get_category_by_id()` (pymapia.PyMapia method), 9
`get_nearest_place()` (pymapia.PyMapia method), 6
`get_place_by_area()` (pymapia.PyMapia method), 6
`get_place_by_id()` (pymapia.PyMapia method), 5
`get_street_by_id()` (pymapia.PyMapia method), 8

P

`PyMapia` (class in pymapia), 5
`pymapia` (module), 5

S

`search_place()` (pymapia.PyMapia method), 7

U

`update_place()` (pymapia.PyMapia method), 8