
pytrojmiastopl Documentation

Release 0.0.3

Arkadiusz Klein

Sep 04, 2017

Contents:

1	Introduction	1
1.1	Scraping category data	1
1.2	Scraping offer data	2
2	Category methods	3
3	Offer methods	5
4	Utils methods	9
5	Indices and tables	11
	Python Module Index	13

pytrojmiastopl supplies two methods to scrape data from www.ogloszenia.trojmiasto.pl website

Scraping category data

This method scrapes available offer urls from trojmiasto.pl search results with parameters

`trojmiastopl.category.get_category(category, region=None, **filters)`

Parses available offer urls from given category from every page

Parameters

- **category** – Search category
- **region** – Search region
- **filters** – Dictionary with additional filters. Following example dictionary contains every possible filter

with examples of it's values.

Example

```
input_dict = { "offer_type": "Mieszkanie", # offer type. See :meth:'utils.decode_type' for reference
               "cena[]": (300, None), # price (from, to). None if you don't want to pass one of arguments
               "kaucja[]": (100,1000), # deposit
               "cena_za_m2[]": (5, 100), # price/surface
               "powierzchnia[]": (23, 300), # surface
               "l_pokoi[]": (2, 5), # desired number of rooms
               "pietro[]": (-1, 6), # desired floor, enum: from 1 to 49 and -1 (ground floor)
               "l_pieter[]": (1, 10), # desired total number of floors in building
               "rok_budowy[]": (2003, 2017), # date of built
               "data_wprow": "1d" # date of adding offer. Available: 1d - today, 3d - 3 days ago, 1w - one week ago,
               # 3w - 3 weeks ago
            }
```

Returns List of all offers for given parameters

Return type list

It can be used like this:

```
input_dict = {"cena[]": (300, None)}
parsed_urls = trojmiastopl.category.get_category("nieruchomosci-mam-do-wynajecia",
↪ "Gdańsk", **input_dict)
```

The above code will put a list of urls containing all the apartments found in the given category into the `parsed_url` variable

Scraping offer data

This method scrapes all offer details from

It can be used like this:

```
descriptions = trojmiastopl.offer.get_descriptions(parsed_urls)
```

The above code will put a list of offer details for each offer url provided in `parsed_urls` into the `descriptions` variable

Category methods

`trojmiastopl.category.get_category(category, region=None, **filters)`

Parses available offer urls from given category from every page

Parameters

- **category** – Search category
- **region** – Search region
- **filters** – Dictionary with additional filters. Following example dictionary contains every possible filter

with examples of it's values.

Example

```
input_dict = { "offer_type": "Mieszkanie", # offer type. See :meth:'utils.decode_type' for reference "cena[]":
               (300, None), # price (from, to). None if you don't want to pass one of arguments "kaucja[]": (100,1000), #
               deposit "cena_za_m2[]": (5, 100), # price/surface "powierzchnia[]": (23, 300), # surface "l_pokoi[]": (2,
               5), # desired number of rooms "pietro[]": (-1, 6), # desired floor, enum: from 1 to 49 and -1 (ground floor)
               "l_pieter[]": (1, 10), # desired total number of floors in building "rok_budowy[]": (2003, 2017), # date of
               built "data_wprow": "1d" # date of adding offer. Available: 1d - today, 3d - 3 days ago, 1w - one week
               ago,
               # 3w - 3 weeks ago
            }
```

Returns List of all offers for given parameters

Return type list

`trojmiastopl.category.get_offers_for_page(category, region, page, **filters)`

Parses offers for one specific page of given category with filters.

Parameters

- **category** (*str*) – Search category

- **region** (*str*) – Search region
- **page** (*int*) – Page number
- **filters** (*dict*) – See :meth category.get_category for reference

Returns List of all offers for given page and parameters

Return type list

`trojmiastopl.category.get_page_count (markup)`

Reads total page number from trojmiastopl.pl search page

Parameters **markup** (*str*) – trojmiastopl.pl search page markup

Returns Total page number

Return type int

Except If no page number was found - there is just one page.

`trojmiastopl.category.get_page_count_for_filters (category, region=None, **filters)`

Reads total page number for given search filters

Parameters

- **category** (*str*) – Search category
- **region** (*str*) – Search region
- **filters** (*dict*) – See :meth category.get_category for reference

Returns Total page number

Return type int

Except If no page number was found - there is just one page.

`trojmiastopl.category.parse_available_offers (markup)`

Collects all offer links on search page markup

Parameters **markup** (*str*) – Search page markup

Returns Links to offer on given search page

Return type list

`trojmiastopl.category.parse_offer_url (markup)`

Searches for offer links in markup

Parameters **markup** (*str*) – Search page markup

Returns Url with offer

Return type str

`trojmiastopl.offer.get_additional_information(offer_markup)`

Searches for additional info and heating type

Parameters `offer_markup` (*str*) – Class “sidebar” from offer page markup

Returns Additional info with optional heating type

Return type dict

`trojmiastopl.offer.get_apartment_type(offer_markup)`

Searches for apartment type in offer markup

Parameters `offer_markup` (*str*) – Class “sidebar” from offer page markup

Returns Apartment type

Return type str

`trojmiastopl.offer.get_available_from(offer_markup)`

Searches for available from in offer markup

Parameters `offer_markup` (*str*) – Class “sidebar” from offer page markup

Returns Available from or None if there is no information

Return type str, None

`trojmiastopl.offer.get_furnished(offer_markup)`

Searches if offer is marked as furnished or not

Parameters `offer_markup` (*str*) – Class “sidebar” from offer page markup

Returns Information is offer furnished

Return type bool

Except If there is no information if offer is furnished it will return None

`trojmiastopl.offer.get_img_url(offer_markup)`

Searches for images in offer markup

Parameters `offer_markup` (*str*) – Id “gallery” from offer page markup

Returns Images of offer in list

Return type list

`trojmiastopl.offer.get_month_num_for_string` (*value*)

Map for polish month names

Parameters `value` (*str*) – Month value

Returns Month number

Return type int

`trojmiastopl.offer.get_surface` (*offer_markup*)

Searches for surface in offer markup

Parameters `offer_markup` (*str*) – Class “sidebar” from offer page markup

Returns Surface or None if there is no surface

Return type float, None

Except When there is no offer surface it will return None

`trojmiastopl.offer.get_title` (*offer_markup*)

Searches for offer title on offer page

Parameters `offer_markup` (*str*) – Class “title-wrap” from offer page markup

Returns Title of offer or None if there is no title

Return type str, None

Except Returns None when couldn’t find title of offer page.

`trojmiastopl.offer.parse_date_to_timestamp` (*date*)

Parses string date to unix timestamp

Parameters `date` (*str*) – Date

Returns Unix timestamp

Return type int

`trojmiastopl.offer.parse_dates_and_id` (*offer_markup*)

Searches for date of creating and date of last update of an offer. Additionally parses offer id number.

Parameters `offer_markup` (*str*) – Class “sidebar” from offer page markup

Returns Date added and date updated if found and offer id (id, added, updated)

Return type dict

`trojmiastopl.offer.parse_description` (*description_markup*)

Searches for offer description

Parameters `description_markup` (*str*) – Class “ogl-description” from offer page markup

Returns Offer description

Return type str

`trojmiastopl.offer.parse_flat_data` (*offer_markup*)

Parses flat data from sidebar

Parameters `offer_markup` (*str*) – Class “sidebar” from offer page markup

Returns Information about price, deposit, floor, number of rooms, date of built and total count of floors in building :rtype: dict

`trojmiastopl.offer.parse_offer(url)`

Parses data from offer page url

Parameters `url` (*str*) – Url of current offer page

Returns Dictionary with all offer details

Return type dict

Except If there is no offer title anymore - offer got deleted.

`trojmiastopl.offer.parse_poster_name(contact_markup)`

Parses poster name

Parameters `contact_markup` (*str*) – Class “contact-box” from offer page markup

Returns Poster name

Return type str

`trojmiastopl.offer.parse_region(offer_markup)`

Parses region information

Parameters `offer_markup` (*str*) – Class “sidebar” from offer page markup

Returns Region of offer

Return type dict

CHAPTER 4

Utils methods

`trojmiastopl.utils.decode_category_name(category)`

Decodes category name to it's value

Parameters `category` (*str*) – Category name

Returns Category number

Return type int

`trojmiastopl.utils.decode_type(filter_value)`

Decodes offer type name to it's value

List of available options and it's translation can be found bellow.

Parameters `filter_value` (*str*) – One of available type names

Returns Int value for POST variable

Return type int

`trojmiastopl.utils.get_content_for_url(url)`

Connects with given url

If environmental variable DEBUG is True it will cache response for url in /var/temp directory

Parameters `url` (*str*) – Website url

Returns Response for requested url

`trojmiastopl.utils.get_url(category, region=None, **filters)`

Creates url for given parameters

Parameters

- **category** (*str*) – Search category
- **region** (*str*) – Search region
- **filters** (*dict*) – Dictionary with additional filters. See :meth:'trojmiastopl.get_category' for reference

Returns Url for given parameters

Return type str

`trojmiastopl.utils.get_url_for_filters(payload)`

Parses url from trojmiasto.pl search engine using POST method for given payload of data

Parameters `payload` (*tuple*) – Tuple of tuples containing POST key and argument

Returns Url generated by trojmiasto.pl search engine

Return type str

CHAPTER 5

Indices and tables

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

t

`trojmiastopl.category`, 3
`trojmiastopl.offer`, 5
`trojmiastopl.utils`, 9

D

`decode_category_name()` (in module `trojmiasopl.utils`), 9
`decode_type()` (in module `trojmiasopl.utils`), 9

G

`get_additional_information()` (in module `trojmiasopl.offer`), 5
`get_apartment_type()` (in module `trojmiasopl.offer`), 5
`get_available_from()` (in module `trojmiasopl.offer`), 5
`get_category()` (in module `trojmiasopl.category`), 1, 3
`get_content_for_url()` (in module `trojmiasopl.utils`), 9
`get_furnished()` (in module `trojmiasopl.offer`), 5
`get_img_url()` (in module `trojmiasopl.offer`), 5
`get_month_num_for_string()` (in module `trojmiasopl.offer`), 6
`get_offers_for_page()` (in module `trojmiasopl.category`), 3
`get_page_count()` (in module `trojmiasopl.category`), 4
`get_page_count_for_filters()` (in module `trojmiasopl.category`), 4
`get_surface()` (in module `trojmiasopl.offer`), 6
`get_title()` (in module `trojmiasopl.offer`), 6
`get_url()` (in module `trojmiasopl.utils`), 9
`get_url_for_filters()` (in module `trojmiasopl.utils`), 10

P

`parse_available_offers()` (in module `trojmiasopl.category`), 4
`parse_date_to_timestamp()` (in module `trojmiasopl.offer`), 6
`parse_dates_and_id()` (in module `trojmiasopl.offer`), 6
`parse_description()` (in module `trojmiasopl.offer`), 6
`parse_flat_data()` (in module `trojmiasopl.offer`), 6
`parse_offer()` (in module `trojmiasopl.offer`), 7
`parse_offer_url()` (in module `trojmiasopl.category`), 4
`parse_poster_name()` (in module `trojmiasopl.offer`), 7
`parse_region()` (in module `trojmiasopl.offer`), 7

T

`trojmiasopl.category` (module), 3
`trojmiasopl.offer` (module), 5
`trojmiasopl.utils` (module), 9