
pyolx Documentation

Release 0.0.3

Arkadiusz Klein

Aug 03, 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

pyolx supplies two methods to scrape data from www.olx.pl website

Scraping category data

This method scrapes available offer urls from OLX search results with parameters

```
olx.category.get_category(main_category, sub_category, detail_category, region, **filters)
```

Parses available offer urls from given category from every page

Parameters

- **main_category** – Main category
- **sub_category** – Sub category
- **detail_category** – Detail category
- **region** – Region of search
- **filters** – Dictionary with additional filters. Following example dictionary contains every possible filter

with examples of it's values.

Example

```
input_dict = { "[filter_float_price:from]": 2000, # minimal price "[filter_float_price:to]": 3000, # maximal price "[filter_enum_floor_select][0]": 3, # desired floor, enum: from -1 to 11 (10 and more) and 17 (attic) "[filter_enum_furniture][0]": True, # furnished or unfurnished offer "[filter_enum_builttype][0]": "blok", # valid build types: # blok, kamienica, szeregowiec, apartamentowiec, wolnostojacy, loft "[filter_float_m:from]": 25, # minimal surface "[filter_float_m:to]": 50, # maximal surface "[filter_enum_rooms][0]": 2 # desired number of rooms, enum: from 1 to 4 (4 and more) }
```

Returns List of all offers for given parameters

Return type list

It can be used like this:

```
input_dict = {'[filter_float_price:from]': 2000}
parsed_urls = olx.category.get_category("nieruchomosci", "mieszkania", "wynajem",
↪ "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

`olx.offer.get_descriptions(parsed_urls)`

Parses details of offers in category

Parameters `parsed_urls` (*list*) – List of offers urls

Returns List of details of offers

Return type list

Except If this offer is not available anymore

It can be used like this:

```
descriptions = olx.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

`olx.category.get_category(main_category, sub_category, detail_category, region, **filters)`

Parses available offer urls from given category from every page

Parameters

- **main_category** – Main category
- **sub_category** – Sub category
- **detail_category** – Detail category
- **region** – Region of search
- **filters** – Dictionary with additional filters. Following example dictionary contains every possible filter

with examples of it's values.

Example

```
input_dict = { "[filter_float_price:from]": 2000, # minimal price "[filter_float_price:to]": 3000, # maximal price "[filter_enum_floor_select][0]": 3, # desired floor, enum: from -1 to 11 (10 and more) and 17 (attic) "[filter_enum_furniture][0]": True, # furnished or unfurnished offer "[filter_enum_builttype][0]": "blok", # valid build types: # blok, kamienica, szeregowiec, apartamentowiec, wolnostojacy, loft "[filter_float_m:from]": 25, # minimal surface "[filter_float_m:to]": 50, # maximal surface "[filter_enum_rooms][0]": 2 # desired number of rooms, enum: from 1 to 4 (4 and more) }
```

Returns List of all offers for given parameters

Return type list

`olx.category.get_offers_for_page(main_category, sub_category, detail_category, region, page, **filters)`

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

Parameters

- **main_category** (*str*) – Main category
- **sub_category** (*str*) – Sub category
- **detail_category** (*str*) – Detail category
- **region** (*str*) – Region of search
- **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

`olx.category.get_page_count` (*markup*)

Reads total page number from OLX search page

Parameters **markup** (*str*) – OLX search page markup

Returns Total page number extracted from js script

Return type int

`olx.category.get_page_count_for_filters` (*main_category*, *sub_category*, *detail_category*, *region*, ***filters*)

Reads total page number for given search filters

Parameters

- **main_category** (*str*) – Main category
- **sub_category** (*str*) – Sub category
- **detail_category** (*str*) – Detail category
- **region** (*str*) – Region of search
- **filters** – See :meth category.get_category for reference

Returns Total page number

Return type int

`olx.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

`olx.category.parse_offer_url` (*markup*)

Searches for offer links in markup

Offer links on OLX are in class “linkWithHash”. Only www.olx.pl domain is whitelisted.

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

Returns Url with offer

Return type str

`olx.offer.get_additional_rent (offer_markup)`

Searches for additional rental costs

Parameters `offer_markup` (*str*) –

Returns Additional rent

Return type int

`olx.offer.get_date_added (offer_markup)`

Searches of date of adding offer

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

Returns Date of adding offer

Return type str

`olx.offer.get_descriptions (parsed_urls)`

Parses details of offers in category

Parameters `parsed_urls` (*list*) – List of offers urls

Returns List of details of offers

Return type list

Except If this offer is not available anymore

`olx.offer.get_gps (offer_markup)`

Searches for gps coordinates (latitude and longitude)

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

Returns Tuple of gps coordinates

Return type tuple

`olx.offer.get_img_url (offer_markup)`

Searches for images in offer markup

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

Returns Images of offer in list

Return type list

`olx.offer.get_poster_name(offer_markup)`

Searches for poster name

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

Returns Poster name

Return type str

`olx.offer.get_surface(offer_markup)`

Searches for surface in offer markup

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

Returns Surface

Return type float

Except When there is no offer surface it will return None

`olx.offer.get_title(offer_markup)`

Searches for offer title on offer page

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

Returns Title of offer

Return type str

`olx.offer.parse_description(offer_markup)`

Searches for description if offer markup

Parameters `offer_markup` (*str*) – Body from offer page markup

Returns Description of offer

Return type str

`olx.offer.parse_flat_data(offer_markup)`

Parses flat data from script of Google Tag Manager

Data includes if offer private or business, number of floor, number of rooms, built type and furniture.

Parameters `offer_markup` (*str*) – Body from offer page markup

Returns Dictionary of flat data

Return type dict

`olx.offer.parse_offer(markup, url)`

Parses data from offer page markup

Parameters

- **markup** (*str*) – Offer page markup
- **url** (*str*) – Url of current offer page

Returns Dictionary with all offer details

Return type dict

`olx.offer.parse_price(offer_markup)`

Searches for price on offer page

Parameters `offer_markup` (*str*) – Head from offer page

Returns Tuple of int price and it's currency

Return type tuple

`olx.offer.parse_region(offer_markup)`

Parses region information :param offer_markup: :return:

CHAPTER 4

Utils methods

`olx.utils.city_name(city)`

Creates valid OLX url city name

OLX city name can't include polish characters, upper case letters. It also should replace white spaces with dashes.

Parameters `city` (*str*) – City name not in OLX url format

Returns Valid OLX url city name

Return type `str`

Example

```
>> city_name("Ruda Śląska") "ruda-slaska"
```

`olx.utils.flatten(container)`

Flatten a list

Parameters `container` (*list*) – list with nested lists

Returns list with elements that were nested in container

Return type `list`

`olx.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

`olx.utils.get_search_filter(filter_name, filter_value)`

Generates url search filter

Parameters

- **filter_name** (*str*) – Filter name in OLX format. See `:meth:'olx.get_category'` for reference

- **filter_value** – Correct value for filter

Returns Percent-encoded url search filter

:rtype str

Example

```
>> get_search_filter([filter_float_price:from], 2000) "search%5Bfilter_float_price%3Afrom%5D=2000"
```

`olx.utils.get_url(main_category, sub_category, detail_category, region, page=None, **filters)`

Creates url for given parameters

Parameters

- **main_category** (*str*) – Main category
- **sub_category** (*str*) – Sub category
- **detail_category** (*str*) – Detail category
- **region** (*str*) – Region of search
- **page** (*int*) – Page number
- **filters** (*dict*) – Dictionary with additional filters. See :meth:'olx.get_category' for reference

Returns Url for given parameters

Return type str

`olx.utils.replace_all(text, input_dict)`

Replace specific strings in string

Parameters

- **text** (*str*) – string with strings to be replaced
- **input_dict** (*dict*) – dictionary with elements in format string: string to be replaced with

Returns String with replaced strings

Return type str

CHAPTER 5

Indices and tables

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

O

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

C

`city_name()` (in module `olx.utils`), 9

F

`flatten()` (in module `olx.utils`), 9

G

`get_additional_rent()` (in module `olx.offer`), 5

`get_category()` (in module `olx.category`), 1, 3

`get_content_for_url()` (in module `olx.utils`), 9

`get_date_added()` (in module `olx.offer`), 5

`get_descriptions()` (in module `olx.offer`), 2, 5

`get_gps()` (in module `olx.offer`), 5

`get_img_url()` (in module `olx.offer`), 5

`get_offers_for_page()` (in module `olx.category`), 3

`get_page_count()` (in module `olx.category`), 4

`get_page_count_for_filters()` (in module `olx.category`), 4

`get_poster_name()` (in module `olx.offer`), 6

`get_search_filter()` (in module `olx.utils`), 9

`get_surface()` (in module `olx.offer`), 6

`get_title()` (in module `olx.offer`), 6

`get_url()` (in module `olx.utils`), 10

O

`olx.category` (module), 3

`olx.offer` (module), 5

`olx.utils` (module), 9

P

`parse_available_offers()` (in module `olx.category`), 4

`parse_description()` (in module `olx.offer`), 6

`parse_flat_data()` (in module `olx.offer`), 6

`parse_offer()` (in module `olx.offer`), 6

`parse_offer_url()` (in module `olx.category`), 4

`parse_price()` (in module `olx.offer`), 6

`parse_region()` (in module `olx.offer`), 7

R

`replace_all()` (in module `olx.utils`), 10