pyolx Documentation

Release 0.0.3

Arkadiusz Klein

Sep 26, 2020

Contents:

1 Introduction **1**
- 1.1 Scraping category data . . . 1
- 1.2 Scraping offer data . . . 2

2 Category methods **3**

3 Offer methods **7**

4 Utils methods **11**

5 Indices and tables **13**

Python Module Index **15**

Index **17**

CHAPTER 1

Introduction

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

1.1 Scraping category data

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

olx.category.**get_category**(*main_category=None*, *sub_category=None*, *detail_category=None*, *region=None*, *search_query=None*, *url=None*, ***filters*)

Parses available offer urls from given category from every page

Parameters

- **url** – User defined url for OLX page with offers. It overrides category parameters and applies search filters.
- **main_category** – Main category
- **sub_category** – Sub category
- **detail_category** – Detail category
- **region** – Region of search
- **search_query** – Additional search query
- **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) "today": True, # Should search for offer posted today? "yesterday": True, # Should search for offers posted yesterday?

}

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

1.2 Scraping offer data

This method scrapes all offer details from

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

CHAPTER 2

Category methods

olx.category.**get_category**(*main_category=None*, *sub_category=None*, *detail_category=None*, *region=None*, *search_query=None*, *url=None*, ***filters*)

Parses available offer urls from given category from every page

Parameters

- **url** – User defined url for OLX page with offers. It overrides category parameters and applies search filters.
- **main_category** – Main category
- **sub_category** – Sub category
- **detail_category** – Detail category
- **region** – Region of search
- **search_query** – Additional search query
- **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) "today": True, # Should search for offer posted today? "yesterday": True, # Should search for offers posted yesterday?

}

Returns List of all offers for given parameters

Return type list

olx.category.**get_offers_for_page**(*page*, *main_category=None*, *sub_category=None*, *detail_category=None*, *region=None*, *search_query=None*, *url=None*, ***filters*)

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

Parameters

- **page** (*int*) – Page number
- **url** (*str, None*) – User defined url for OLX page with offers. It overrides category parameters and applies search filters.
- **main_category** (*str, None*) – Main category
- **sub_category** (*str, None*) – Sub category
- **detail_category** (*str, None*) – Detail category
- **region** (*str, None*) – Region of search
- **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=None*, *sub_category=None*, *detail_category=None*, *region=None*, *search_query=None*, *url=None*, ***filters*)

Reads total page number for given search filters

Parameters

- **url** (*str, None*) – User defined url for OLX page with offers. It overrides category parameters and applies search filters.
- **main_category** (*str, None*) – Main category
- **sub_category** (*str, None*) – Sub category
- **detail_category** (*str, None*) – Detail category
- **region** (*str, None*) – Region of search
- **search_query** (*str, None*) – Additional search query
- **filters** – See :meth category.get_category for reference

Returns Total page number

Return type int

olx.category.**parse_ads_count**(*markup*)

Reads total number of adds

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

Returns Total ads count from script

Return type int

olx.category.**parse_available_offers**(*markup*, *today=None*, *yesterday=None*)
Collects all offer links on search page markup

Parameters

- **markup** (*str*) – Search page markup
- **today** (*object*) – Should search for offers posted yesterday?
- **yesterday** (*object*) – Should search for offers posted yesterday?

Returns Links to offer on given search page

Return type list

olx.category.**parse_offer_url**(*markup*, *today=None*, *yesterday=None*)
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
- **today** (*object*) – Should search for offers posted yesterday?
- **yesterday** (*object*) – Should search for offers posted yesterday?

Returns Url with offer

Return type str

CHAPTER 3

Offer methods

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 Unix timestamp

Return type int

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_gpt_script**(*offer_markup*)
Parses data from script of Google Tag Manager

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

Returns GPT dict data

Return type dict

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_month_num_for_string**(*value*)
Map for polish month names

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

Returns Month number

Return type int

olx.offer.**get_poster_name**(*offer_markup*)
Searches for poster name

Parameters **offer_markup** (*str*) – Class "offerbody" from offer page markup

Returns Poster name or None if poster name was not found (offer is outdated)

Return type str, None

Except Poster name not found

olx.offer.**get_surface**(*offer_markup*)
Searches for surface in offer markup

Parameters **offer_markup** (*str*) – Class "offerbody" 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

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, None

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*, *data_dict*)
Parses flat data

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
- **data_dict** (*dict*) – Dict with GPT script data

Returns Dictionary of flat data

Return type dict

olx.offer.**parse_offer**(*url*)
Parses data from offer page url

Parameters

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

Returns Dictionary with all offer details or None if offer is not available anymore

Return type dict, None

olx.offer.**parse_region**(*offer_markup*)
Parses region information

Parameters **offer_markup** (*str*) – Class "offerbody" from offer page markup

Returns Region of offer

Return type list

olx.offer.**parse_tracking_data**(*offer_markup*)
Parses price and add_id from OLX tracking data script

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

Returns Tuple of int price and it's currency or None if this offer page got deleted

Return type tuple, None

Except This offer page got deleted and has no tracking script.

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.**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=None*, *sub_category=None*, *detail_category=None*, *region=None*, *search_query=None*, *page=None*, *user_url=None*, ***filters*)

Creates url for given parameters

Parameters

- **user_url** – User defined OLX search url
- **main_category** (*str, None*) – Main category
- **sub_category** (*str, None*) – Sub category
- **detail_category** (*str, None*) – Detail category
- **region** (*str, None*) – Region of search
- **search_query** (*str*) – Search query string
- **page** (*int, None*) – Page number
- **filters** (*dict*) – Dictionary with additional filters. See :meth:’olx.get_category’ for reference

Returns Url for given parameters

Return type str

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Python Module Index

o

olx.category, 3
olx.offer, 7
olx.utils, 11

Index

C

city_name() (*in module olx.utils*), 11

G

get_additional_rent() (*in module olx.offer*), 7
get_category() (*in module olx.category*), 1, 3
get_content_for_url() (*in module olx.utils*), 11
get_date_added() (*in module olx.offer*), 7
get_gps() (*in module olx.offer*), 7
get_gpt_script() (*in module olx.offer*), 7
get_img_url() (*in module olx.offer*), 7
get_month_num_for_string() (*in module olx.offer*), 8
get_offers_for_page() (*in module olx.category*), 4
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*), 8
get_search_filter() (*in module olx.utils*), 11
get_surface() (*in module olx.offer*), 8
get_title() (*in module olx.offer*), 8
get_url() (*in module olx.utils*), 12

O

olx.category (*module*), 3
olx.offer (*module*), 7
olx.utils (*module*), 11

P

parse_ads_count() (*in module olx.category*), 4
parse_available_offers() (*in module olx.category*), 5
parse_description() (*in module olx.offer*), 8
parse_flat_data() (*in module olx.offer*), 8
parse_offer() (*in module olx.offer*), 8
parse_offer_url() (*in module olx.category*), 5
parse_region() (*in module olx.offer*), 9
parse_tracking_data() (*in module olx.offer*), 9