
pyotodom Documentation

Release 0.0.2

Jan Baryła

Nov 13, 2017

Contents

1	Introduction	3
1.1	Scraping category data	3
1.2	Scraping offer data	5
2	Category methods	7
3	Offer methods	11
4	Utils methods	15
	Python Module Index	17

Contents:

pyotodom supplies two methods that can be used to scrape data from OtoDom. They are designed to work in tandem, but they can also be used separately.

1.1 Scraping category data

The following method should be used to scrape all the offers compliant with the supplied search parameters

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

Scrape OtoDom search results based on supplied parameters.

Parameters

- **main_category** – “wynajem” or “sprzedaz”, should not be empty
- **detail_category** – “mieszkanie”, “dom”, “pokoj”, “dzialka”, “lokal”, “haleimagazyny”, “garaz”, or empty string for any
- **region** – a string that contains the region name. Districts, cities and voivodeships are supported. The exact location is established using OtoDom’s API, just as it would happen when typing something into the search bar. Empty string returns results for the whole country. Will be ignored if either ‘city’, ‘region’, ‘[district_id]’ or ‘[street_id]’ is present in the filters.
- **filters** – the following dict contains every possible filter with examples of its values, but can be empty:

```
input_dict = {
    '[dist]': 0, # distance from region
    '[filter_float_price:from]': 0, # minimal price
    '[filter_float_price:to]': 0, # maximal price
    '[filter_float_price_per_m:from]': 0 # maximal price per square meter, only
    ↳used for apartments for sale
    '[filter_float_price_per_m:to]': 0 # minimal price per square meter, only
    ↳used for apartments for sale
```

```

    '[filter_enum_market][]': [primary, secondary] # enum: primary, secondary
    '[filter_enum_building_material][]': [] # enum: brick, wood, breezeblock,
    ↳hydroton, concrete_plate,
        concrete, silikat, cellular_concrete, reinforced_concrete, other, only
    ↳used for apartments for sale
    '[filter_float_m:from]': 0, # minimal surface
    '[filter_float_m:to]': 0, # maximal surface
    '[filter_enum_rooms_num][]': '1', # number of rooms, enum: from "1" to "10",
    ↳or "more"
    '[private_business]': 'private', # poster type, enum: private, business
    '[open_day]': 0, # whether or not the poster organises an open day
    '[exclusive_offer]': 0, # whether or not the offer is otodom exclusive
    '[filter_enum_rent_to_students][]': 0, # whether or not the offer is aimed
    ↳for students, only used for
        apartments for rent
    '[filter_enum_floor_no][]': 'floor_1', # enum: cellar, ground_floor, floor_1-
    ↳floor_10, floor_higher_10,
        garret
    '[filter_float_building_floors_num:from]': 1, # minimal number of floors in
    ↳the building
    '[filter_float_building_floors_num:to]': 1, # maximal number of floors in
    ↳the building
    '[building_type]': 'blok', # enum: blok, w-kamienicy, dom-wolnostojacy, plomba,
    ↳szeregowiec,
        apartamentowiec, loft
    '[filter_enum_heating][]': 'urban', # enum: urban, gas, tiled_stove,
    ↳electrical, boiler_room, other
    '[filter_float_build_year:from]': 1980, # minimal year the building was
    ↳built in
    '[filter_float_build_year:to]': 2016, # maximal year the building was built
    ↳in
    '[filter_enum_extras_types][]': ['balcony', 'basement'], # enum: balcony,
    ↳usable_room, garage, basement,
        garden, terrace, lift, two_storey, separate_kitchen, air_conditioning,
    ↳non_smokers_only
    '[filter_enum_media_types][]': ['internet', 'phone'], # enum: internet,
    ↳cable-television, phone
    '[free_from]': 'from_now', # when will it be possible to move in, enum: from
    ↳now, 30, 90
    '[created_since]': 1, # when was the offer posted on otodom in days, enum: 1,
    ↳3, 7, 14
    '[id]': 48326376, # otodom offer ID, found at the very bottom of each offer
    '[description_fragment]': 'wygodne', # the resulting offers' descriptions must
    ↳contain this string
    '[photos]': 0, # whether or not the offer contains photos
    '[movie]': 0, # whether or not the offer contains video
    '[walkaround_3dview]': 0 # whether or not the offer contains a walkaround 3D
    ↳view
    'city': # lowercase, no diacritics, '-' instead of spaces, _city_id at the
    ↳end
    'voivodeship': # lowercase, no diacritics, '-' instead of spaces
    '[district_id]': from otodom API
    '[street_id]': from otodom API
}

```

Return type list of dict(string, string)

Returns Each of the dictionaries contains the following fields:


```
'detail_url' - a link to the offer
'offer_id' - the internal otodom's offer ID, not to be mistaken with the '[id]'
↳field from the input_dict
'poster' - a piece of information about the poster. Could either be a name of the
↳agency or "Oferta prywatna"
```

It can be used like this:

```
input_dict = {'[filter_float_price:to]': 1100}
parsed_category = scrape.category.get_category("wynajem", "mieszkanie", "gda",
↳**input_dict)
```

The above code will put a list of dictionaries(string, string) containing all the apartments found in the given category (apartments for rent, in a region starting with “gda”, cheaper than 1100 PLN) into the parsed_category variable

1.2 Scraping offer data

The following method should be used to scrape all the information about an offer located under the given string. Context is used for phone number scraping. The corresponding field will be empty if it's not provided.

`otodom.offer.get_offer_information(url, context=None)`

Scrape detailed information about an OtoDom offer.

Parameters

- **url** – a string containing a link to the offer
- **context** – a dictionary(string, string) taken straight from the `scrape.category.get_category()`

Returns A dictionary containing the scraped offer details

It can be used like this:

```
offer_details = []
for offer in parsed_category:
    offer_details.append(get_offer_information(offer['detail_url'], context=offer))
```

The above code will populate the offer_details list with all the information about apartments found in parsed_category

Category methods

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

Scrape OtoDom search results based on supplied parameters.

Parameters

- **main_category** – “wynajem” or “sprzedaz”, should not be empty
- **detail_category** – “mieszkanie”, “dom”, “pokoj”, “dzialka”, “lokal”, “haleimagazyny”, “garaz”, or empty string for any
- **region** – a string that contains the region name. Districts, cities and voivodeships are supported. The exact location is established using OtoDom’s API, just as it would happen when typing something into the search bar. Empty string returns results for the whole country. Will be ignored if either ‘city’, ‘region’, ‘[district_id]’ or ‘[street_id]’ is present in the filters.
- **filters** – the following dict contains every possible filter with examples of its values, but can be empty:

```
input_dict = {
    '[dist]': 0, # distance from region
    '[filter_float_price:from]': 0, # minimal price
    '[filter_float_price:to]': 0, # maximal price
    '[filter_float_price_per_m:from]': 0 # maximal price per square meter, only
    ↳used for apartments for sale
    '[filter_float_price_per_m:to]': 0 # minimal price per square meter, only
    ↳used for apartments for sale
    '[filter_enum_market][]': [primary, secondary] # enum: primary, secondary
    '[filter_enum_building_material][]': [] # enum: brick, wood, breezeblock,
    ↳hydroton, concrete_plate,
    concrete, silikat, cellular_concrete, reinforced_concrete, other, only
    ↳used for apartments for sale
    '[filter_float_m:from]': 0, # minimal surface
    '[filter_float_m:to]': 0, # maximal surface
    '[filter_enum_rooms_num][]': '1', # number of rooms, enum: from "1" to "10",
    ↳or "more"
```

```

    '[private_business]': 'private', # poster type, enum: private, business
    '[open_day]': 0, # whether or not the poster organises an open day
    '[exclusive_offer]': 0, # whether or not the offer is otodom exclusive
    '[filter_enum_rent_to_students]': 0, # whether or not the offer is aimed
    ↪ for students, only used for
        apartments for rent
    '[filter_enum_floor_no]': 'floor_1', # enum: cellar, ground_floor, floor_1-
    ↪ floor_10, floor_higher_10,
        garret
    '[filter_float_building_floors_num:from]': 1, # minimal number of floors in
    ↪ the building
    '[filter_float_building_floors_num:to]': 1, # maximal number of floors in
    ↪ the building
    'building_type': 'blok', # enum: blok, w-kamienicy, dom-wolnostojacy, plomba,
    ↪ szeregowiec,
        apartamentowiec, loft
    '[filter_enum_heating]': 'urban', # enum: urban, gas, tiled_stove,
    ↪ electrical, boiler_room, other
    '[filter_float_build_year:from]': 1980, # minimal year the building was
    ↪ built in
    '[filter_float_build_year:to]': 2016, # maximal year the building was built
    ↪ in
    '[filter_enum_extras_types]': ['balcony', 'basement'], # enum: balcony,
    ↪ usable_room, garage, basement,
        garden, terrace, lift, two_storey, separate_kitchen, air_conditioning,
    ↪ non_smokers_only
    '[filter_enum_media_types]': ['internet', 'phone'], # enum: internet,
    ↪ cable-television, phone
    '[free_from]': 'from_now', # when will it be possible to move in, enum: from
    ↪ now, 30, 90
    '[created_since]': 1, # when was the offer posted on otodom in days, enum: 1,
    ↪ 3, 7, 14
    '[id]': 48326376, # otodom offer ID, found at the very bottom of each offer
    'description_fragment': 'wygodne', # the resulting offers' descriptions must
    ↪ contain this string
    '[photos]': 0, # whether or not the offer contains photos
    '[movie]': 0, # whether or not the offer contains video
    '[walkaround_3dview]': 0 # whether or not the offer contains a walkaround 3D
    ↪ view
    'city': # lowercase, no diacritics, '-' instead of spaces, _city_id at the
    ↪ end
    'voivodeship': # lowercase, no diacritics, '-' instead of spaces
    '[district_id]': from otodom API
    '[street_id]': from otodom API
}

```

Return type list of dict(string, string)

Returns Each of the dictionaries contains the following fields:

```

'detail_url' - a link to the offer
'offer_id' - the internal otodom's offer ID, not to be mistaken with the '[id]'.
    ↪ field from the input_dict
'poster' - a piece of information about the poster. Could either be a name of the
    ↪ agency or "Oferta prywatna"

```

otodom.category.get_category_number_of_pages (markup)

A method that returns the maximal page number for a given markup, used for pagination handling.

Parameters `markup` – a `requests.response.content` object

Return type `int`

`otodom.category.get_category_number_of_pages_from_parameters` (*main_category*,
detail_category,
region, ***filters*)

A method to establish the number of pages before actually scraping any data

`otodom.category.get_distinct_category_page` (*page*, *main_category*, *detail_category*, *region*,
***filters*)

A method for scraping just the distinct page of a category

`otodom.category.parse_category_content` (*markup*)

A method for getting a list of all the offers found in the markup.

Parameters `markup` – a `requests.response.content` object

Return type `list(requests.response.content)`

`otodom.category.parse_category_offer` (*offer_markup*)

A method for getting the most important data out of an offer markup.

Parameters `offer_markup` – a `requests.response.content` object

Return type `dict(string, string)`

Returns see the return section of `scrape.category.get_category()` for more information

CHAPTER 3

Offer methods

`otodom.offer.get_month_num_for_string(value)`

Map for polish month names

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

Returns Month number

Return type `int`

`otodom.offer.get_offer_3d_walkaround_link(html_parser)`

This method returns a link to a 3D walkaround view of the apartment.

Parameters `html_parser` – a BeautifulSoup object

Return type `string`

Returns A 3D walkaround view of the apartment

`otodom.offer.get_offer_additional_assets(html_parser)`

This method returns information about the apartment's additional assets.

Parameters `html_parser` – a BeautifulSoup object

Return type `list(string)`

Returns A list containing the additional assets

`otodom.offer.get_offer_address(html_parser)`

This method returns the offer address.

Parameters `html_parser` – a BeautifulSoup object

Return type `string`

Returns The offer address

`otodom.offer.get_offer_apartment_details(html_parser)`

This method returns detailed information about the apartment.

Parameters `html_parser` – a BeautifulSoup object

Return type list(dict)

Returns A list containing dictionaries of details, for example { 'kaucja': 1100 zł }

`otodom.offer.get_offer_description(html_parser)`

This method returns the apartment description.

Parameters `html_parser` – a BeautifulSoup object

Return type string

Returns The apartment description

`otodom.offer.get_offer_details(html_parser)`

This method returns detailed information about the offer.

Parameters `html_parser` – a BeautifulSoup object

Return type list(dict)

Returns A list of dictionaries containing information about the offer

`otodom.offer.get_offer_facebook_description(html_parser)`

This method returns the short standardized description used for the default facebook share message.

Parameters `html_parser` – a BeautifulSoup object

Return type string

Returns The default facebook share message

`otodom.offer.get_offer_floor(html_parser)`

This method returns the floor on which the apartment is located.

Parameters `html_parser` – a BeautifulSoup object

Return type string

Returns The floor number

`otodom.offer.get_offer_geographical_coordinates(html_parser)`

This method returns the geographical coordinates of the apartment.

Parameters `html_parser` – a BeautifulSoup object

Return type tuple(string)

Returns A tuple containing the latitude and longitude of the apartment

`otodom.offer.get_offer_information(url, context=None)`

Scrape detailed information about an OtoDom offer.

Parameters

- **url** – a string containing a link to the offer
- **context** – a dictionary(string, string) taken straight from the `scrape.category.get_category()`

Returns A dictionary containing the scraped offer details

`otodom.offer.get_offer_ninja_pv(html_content)`

This method returns the website's ninjaPV json data as dict.

Parameters `html_content` – a `requests.response.content` object

Return type dict

Returns ninjaPV data

`otodom.offer.get_offer_phone_numbers(offer_id, cookie, csrf_token)`

This method makes a request to the OtoDom API asking for the poster's phone number(s) and returns it.

Parameters

- **offer_id** – string, taken from context, see the return section of `scrape.category.get_category()` for reference
- **cookie** – string, see `scrape.utils.get_cookie_from()` for reference
- **csrf_token** – string, see `scrape.utils.get_csrf_token()` for reference

Return type list(string)

Returns A list of phone numbers as strings (no spaces, no '+48')

`otodom.offer.get_offer_photos_links(html_parser)`

This method returns a list of links to photos of the apartment.

Parameters **html_parser** – a BeautifulSoup object

Return type list(string)

Returns A list of links to photos of the apartment

`otodom.offer.get_offer_poster_name(html_parser)`

This method returns the poster's name (and surname if available).

Parameters **html_parser** – a BeautifulSoup object

Return type string

Returns The poster's name

`otodom.offer.get_offer_title(html_parser)`

This method returns the offer title.

Parameters **html_parser** – a BeautifulSoup object

Return type string

Returns The offer title

`otodom.offer.get_offer_total_floors(html_parser, default_value='')`

This method returns the maximal number of floors in the building.

Parameters **html_parser** – a BeautifulSoup object

Return type string

Returns The maximal floor number

`otodom.offer.get_offer_video_link(html_parser)`

This method returns a link to a video of the apartment.

Parameters **html_parser** – a BeautifulSoup object

Return type string

Returns A link to a video of the apartment

`otodom.offer.parse_available_from(date)`

Parses string date to unix timestamp

Parameters **date** (*str*) – Date

Returns Unix timestamp

Return type int

`otodom.offer.parse_date_to_timestamp(date)`

Parses string date to unix timestamp

Parameters `date` (*str*) – Date

Returns Unix timestamp

Return type int

CHAPTER 4

Utils methods

`otodom.utils.get_cookie_from(response)`

Parameters `response` – a `requests.response` object

Return type string

Returns cookie information as string

`otodom.utils.get_csrf_token(html_content)`

Parameters `html_content` – a `requests.response.content` object

Return type string

Returns the CSRF token as string

`otodom.utils.get_region_from_autosuggest(region_part)`

This method makes a request to the OtoDom api, asking for the best fitting region for the supplied `region_part` string.

Parameters `region_part` – input string, it should be a part of an existing region in Poland, either city, street, district or voivodeship

Return type dict

Returns A dictionary which contents depend on the API response.

`otodom.utils.get_region_from_filters(filters)`

This method does a similiar thing as `scrape.utils.get_region_from_autosuggest()` but instead of calling the API, it uses the data provided in the filters

Parameters `filters` – dict, see `scrape.category.get_category()` for reference

Return type dict

Returns A dictionary which contents depend on the filters content.

`otodom.utils.get_response_for_url(url)`

Parameters `url` – an url, most likely from the `scrape.utils.get_url()` method

Returns a requests.response object

`otodom.utils.get_url(main_category, detail_category, region, ads_per_page='', page=None, **filters)`

This method builds a ready-to-use url based on the input parameters.

Parameters

- **main_category** – see `scrape.category.get_category()` for reference
- **detail_category** – see `scrape.category.get_category()` for reference
- **region** – see `scrape.category.get_category()` for reference
- **ads_per_page** – “?nrAdsPerPage=72” can be used to lower the amount of requests
- **page** – page number
- **filters** – see `scrape.category.get_category()` for reference

Return type string

Returns the url

O

`otodom.category`, [7](#)
`otodom.offer`, [11](#)
`otodom.utils`, [15](#)

G

`get_category()` (in module `otodom.category`), [3](#), [7](#)
`get_category_number_of_pages()` (in module `otodom.category`), [8](#)
`get_category_number_of_pages_from_parameters()` (in module `otodom.category`), [9](#)
`get_cookie_from()` (in module `otodom.utils`), [15](#)
`get_csrf_token()` (in module `otodom.utils`), [15](#)
`get_distinct_category_page()` (in module `otodom.category`), [9](#)
`get_month_num_for_string()` (in module `otodom.offer`), [11](#)
`get_offer_3d_walkaround_link()` (in module `otodom.offer`), [11](#)
`get_offer_additional_assets()` (in module `otodom.offer`), [11](#)
`get_offer_address()` (in module `otodom.offer`), [11](#)
`get_offer_apartment_details()` (in module `otodom.offer`), [11](#)
`get_offer_description()` (in module `otodom.offer`), [12](#)
`get_offer_details()` (in module `otodom.offer`), [12](#)
`get_offer_facebook_description()` (in module `otodom.offer`), [12](#)
`get_offer_floor()` (in module `otodom.offer`), [12](#)
`get_offer_geographical_coordinates()` (in module `otodom.offer`), [12](#)
`get_offer_information()` (in module `otodom.offer`), [5](#), [12](#)
`get_offer_ninja_pv()` (in module `otodom.offer`), [12](#)
`get_offer_phone_numbers()` (in module `otodom.offer`), [12](#)
`get_offer_photos_links()` (in module `otodom.offer`), [13](#)
`get_offer_poster_name()` (in module `otodom.offer`), [13](#)
`get_offer_title()` (in module `otodom.offer`), [13](#)
`get_offer_total_floors()` (in module `otodom.offer`), [13](#)
`get_offer_video_link()` (in module `otodom.offer`), [13](#)
`get_region_from_autosuggest()` (in module `otodom.utils`), [15](#)
`get_region_from_filters()` (in module `otodom.utils`), [15](#)
`get_response_for_url()` (in module `otodom.utils`), [15](#)
`get_url()` (in module `otodom.utils`), [16](#)

O

`otodom.category` (module), [7](#)
`otodom.offer` (module), [11](#)
`otodom.utils` (module), [15](#)

P

`parse_available_from()` (in module `otodom.offer`), [13](#)
`parse_category_content()` (in module `otodom.category`), [9](#)
`parse_category_offer()` (in module `otodom.category`), [9](#)
`parse_date_to_timestamp()` (in module `otodom.offer`), [13](#)