abclinuxuapi

Release 0.4.11

October 25, 2016

Contents

Python API for the http://abclinuxu.cz website.

CHAPTER 1

API

Whole API is splitted into four submodules:

`/api/abclinuxuapi`:

1.1 user submodule

This module is here to allow basic manipulation with users. It allows you to retreive all blogposts for fiven user, upload new concepts, register blog and so on.

class `abclinuxuapi.user.`**`User`**(*username*, *password=None*, *lazy=False*)

Bases: `object`

Class that is used to hold informations about given *username*. You can also command various operations, like get list of all blogs, or add new concept.

`username`

str

`password`

str, default None – Password for logged user.

`logged_in`

bool – Is the user logged in?

Parameters

- **`username`** (`str`) – Users login.
- **`password`** (`str, default None`) – Optional password for given user. This will allow you to upload concepts.
- **`lazy`** (`bool, default False`) – Don't call `lazy_init()` right when the object is created.

`has_blog`

Does the user have registered blog?

`lazy_init`()

Parse additional informations about user. This step require one request to the site.

static `from_user_id`(*user_id*)

Transform *user_id* to instance of `User`.

Returns `User` instance parsed from the *user_id*.

Return type obj

login(*password=None*)

Logs the user in, tests, if the user is really logged.

Parameters **password** (*str, default None*) – Password, overwrites the password set when the object was created.

Raises UserWarning – if there was some error during login.

get_blogposts()

Lists all of users PUBLISHED blogposts. For unpublished, see *get_concepts()*.

Returns sorted (old->new) list of Blogpost objects.

Return type list

get_concepts()

Return all concepts (unpublished blogs).

Returns List of Concept objects.

Return type list

add_concept(*text*, *title*, *ts_of_pub=None*)

Adds new concept into your concepts.

Parameters

- **text** (*str*) – Text of your concept.
- **title** (*str*) – Title of your contept. Do not use HTML in title!
- **ts_of_pub** (*int/float, default None*) – Timestamp of the publication.

Raises UserWarning – if the site is broken or user was logged out.

register_blog(*blog_name*)

Register blog under *blog_name*. Users doesn't have blogs automatically, you have to create them manually.

Raises

- UserWarning – If user already have blog registered.
- ValueError – If it is not possible to register blog for user (see exception message for details).

1.2 blogpost submodule

This module is used to represent all kinds of informations about blogposts.

class abclinuxuapi.blogpost.**Rating**

Bases: *abclinuxuapi.blogpost.Rating*

Container holding informations about rating.

rating

int – Percentual rating of the blogpost.

base

int – How many people voted.

class `abclinuxuapi.blogpost.`**`Tag`**(*tag*, *norm=None*)

Bases: `str`

Each blog may have many tags. This is container for informations about each tag.

`tag`

str – Human readable content of the tag.

`norm`

str – norm machine-readable version of tag.

`url`

class `abclinuxuapi.blogpost.`**`Blogpost`**(*url*, *lazy=True*, ***kwargs*)

Bases: `object`

Informations about blogposts.

`url`

str – Absolute URL of the blogpost.

`uid`

int – Unique identificator of the blogpost.

`title`

str – Tile of the blogpost.

`intro`

str – Perex. This is parsed only when returned from `User`.

`text`

str – Full text of the blogpost.

`tags`

list – List of `Tag` objects.

`rating`

obj – `Rating` object with informations about rating.

`has_tux`

bool – Does this blog have a tux? Only good blogs get tux.

`comments`

list – List of `Comment` objects. Not used until `pull()` is called, or *lazy* parameter of `__init__()` is set to `True`.

`comments_n`

int – Number of comments. This information is in some cases known before the blog is parsed, just from perex.

`readed`

int – How many times was the blog readed?

`object_ts`

int – Timestamp of the creation of this object.

`created_ts`

int – Timestamp of the creation of the blogpost.

`last_modified_ts`

int – Timestamp of the last modification of blogpost.

Parameters

- **url** (*str*) – Url of the blogpost.
- **lazy** (*bool, default True*) – True == don't call *pull()* right now.

static **from_html** (*html*, *lazy=True*)

Convert HTML string to *Blogpost* instance.

Parameters

- **html** (*str*) – Input data.
- **lazy** (*bool, default True*) – Be lazy (don't pull data by yourself from the site). Call *pull()* for active download of all required informations.

Returns *Blogpost* instance.

Return type obj

pull ()

Download page with blogpost. Parse text, comments and everything else.

Until this is called, following attributes are not known/parsed:

- *text*
- *tags*
- *has_tux*
- *comments*
- *last_modified_ts*

get_image_urls ()

Get list of links to all images used in this blog.

Returns List of str containing absolute URL of the image.

Return type list

classmethod **possible_tags** ()

Get list of all possible tags which may be set.

Returns List of *Tag* objects.

Return type list

add_tag (*tag*)

Add new tag to the blogpost.

Parameters **tag** (Tag) – *Tag* instance. See *possible_tags* for list of all possible tags.

Raises

- KeyError – In case, that *tag* is not instance of *Tag*.
- ValueError – In case that *uid* is not set.

Returns List of *Tag* objects.

Return type list

remove_tag (*tag*, *throw=False*)

Remove tag from the tags currently assigned to blogpost.

Parameters

- **tag** (Tag) – *Tag* instance. See *possible_tags* for list of all possible tags.

- **throw** (*bool*) – Raise error in case you are trying to remove tag that is not assigned to blogpost.

Raises

- `KeyError` – In case, that *tag* is not instance of *Tag*.
- `IndexError` – In case that you are trying to remove tag which is not assigned to blogpost.
- `ValueError` – In case that *uid* is not set.

Returns List of *Tag* objects.

Return type list

1.3 comment submodule

This module defines the *Comment* class which is used to repesent informations about comments.

class `abclinuxuapi.comment.`**`Comment`**

Bases: `object`

Comment representation.

Note: For registered users, the *username* property contains *real* username, which may differ from what you see, but this allows you to identify the user.

This is because registered users can (and do) change their *visible* usernames anytime they want.

url

str – Absolute URL of the comment.

text

str – Fulltext of the comment.

timestamp

int – Date of the publication as timestamp.

username

str – Username of the poster.

registered

bool – Was the user registered?

censored

bool – Is the comment censored? If so, you will need additional parsing of the comment, which is not yet implemented.

responses

list – List of *Comment* instances responding to this comment.

response_to

obj – Reference to *Comment* to which you are responding. `None` in cases where the object is at the top of the comment tree.

id

Returns – str: Identification of the comment.

static comments_from_html (*html*)

Parse comments in *html*, return list of connected `Comment` instances.

Parameters **html** (*str*) – Webpage for parsing.

Returns List of `Comment` instances linked also into trees using `response_to` and `responses` properties.

Return type list

1.4 concept submodule

Representation and manipulation with concepts. List of concepts can be obtained from logged `User` using `User.get_concepts()` getter.

Note: This module was one of the first modules created here and will require some refactoring.

class abclinuxuapi.concept.**Concept** (*title*, *link*, *session*)

Bases: `object`

title

str – Title of the concept.

link

str – Absolute URL of the concept.

get_content ()

Get content of this Concept.

Returns full HTML UTF-8 encoded text of the concept.

Return type str

add_pic (*opened_file*)

Add picture to the Concept.

Parameters **opened_file** (*file*) – opened file object

list_pics ()

Returns List of URLs to pictures used in this concept.

Return type list

edit (*text*, *title=None*, *date_of_pub=None*)

Edit concept.

Parameters

- **text** (*str*) – New text of the context.
- **title** (*str, default None*) – New title of the concept. If not set, old title is used.
- **date_of_pub** (*str/int, default None*) – Date string in abclinuxu format or timestamp determining when the concept should be automatically published.

Note: *date_of_pub* can be string in format `"%Y-%m-%d %H:%M"`.

And one file with shared functions:

1.5 shared submodule

Functions, objects and properties shared across many classes in this project.

`abclinuxuapi.shared.`**`ABCLINUXU_URL`** **= 'https://www.abclinuxu.cz'**

Base URL of the abclinuxu.

`abclinuxuapi.shared.`**`SESSION`** **= <requests.sessions.Session object>**

Session which is used in non-private requests.

`abclinuxuapi.shared.`**`download`**(*url*, *params=None*, *method='GET'*, *session=None*, *as_text=True*, *data=None*)

Download data from *url* using *method*. Send *params* if defined.

Parameters

- **`url`** (*`str`*) – Absolute URL from which the data will be downloaded.
- **`params`** (*`dict, default None`*) – Send parameters (GET/POST).
- **`data`** (*`dict, default None`*) – Data which will be sent as body of the request.
- **`method`** (*`str, default "GET"`*) – Use this method to send the request.
- **`session`** (*`obj, default None`*) – If `None`, shared *`SESSION`* object is used.
- **`as_text`** (*`bool, default True`*) – Return content as `UTF-8` encoded string. If false `bytes` are returned.

Returns Content depending on the *as_text* attribute.

Return type str/bytes

`abclinuxuapi.shared.`**`first`**(*inp_data*)

Return first element from *inp_data*, or raise StopIteration.

Note: This function was created because it works for generators, lists, iterators, tuples and so on same way, which indexing doesn't.

Also it have smaller cost than list(generator)[0], because it doesn't convert whole *inp_data* to list.

Parameters **`inp_data`** (*`iterable`*) – Any iterable object.

Raises `StopIteration` – When the *inp_data* is blank.

`abclinuxuapi.shared.`**`date_to_timestamp`**(*date*)

Convert abclinuxu *relative* or *absolute* date string in czech words to timestamp.

Parameters **`date`** (*`str`*) – One of many abclinuxu representations of date string.

Returns Timestamp.

Return type int

`abclinuxuapi.shared.`**`date_izolator`**(*lines*)

Return all lines, that looks like it may be date in one of three abclinuxu formats.

Parameters **`lines`** (*`list`*) – List of strings with lines which ma by dates.

Returns List of lines, which looks like they may contain dates.

Return type list

`abclinuxuapi.shared.`**`parse_timestamp`**(*meta*)

Parse numeric timestamp from the date representation.

Parameters **`meta`** (*`str`*) – Meta html from the blogpost body.

Returns Timestamp.

Return type int

`abclinuxuapi.shared.`**`ts_to_concept_date`**(*timestamp*)

Convert numeric *timestamp* into format used by abclinuxu for concepts.

Parameters **`timestamp`** (*`int`*) – Timestamp as float/int.

Returns Converted *timestamp*.

Return type str

`abclinuxuapi.shared.`**`url_context`**(*rel_url*)

Add *rel_url* to the absolute context with *`ABCLINUXU_URL`*.

Parameters **`rel_url`** (*`str`*) – Relative URL.

Returns Absolute URL.

Return type str

`abclinuxuapi.shared.`**`handle_errors`**(*dom*)

Look for error divs in given *dom* tree.

Parameters **dom** (*obj*) – `dhtmlparser.HTMLElement` instance.

Raises `UserWarning` – With content of the error div if the div was found.

`abclinuxuapi.shared.`**`check_error_div`**(*data*, *error_div*)

Dunno about this, it was created long time ago and I have no idea.

`abclinuxuapi.shared.`**`check_error_page`**(*data*)

Handle other, special kind of errors.

CHAPTER 2

Installation

Module is hosted at PYPI, and can be easily installed using PIP:

```
sudo pip install abclinuxuapi
```

CHAPTER 3

Source code

Project is released under MIT license. Source code can be found at GitHub:

- https://github.com/Bystroushaak/abclinuxuapi

3.1 Unittests

A lot of features of the project is tested by unittests. You can run those tests using provided `run_tests.sh` script, which can be found in the root of the project.

If you have any trouble, just add `--pdb` switch at the end of your `run_tests.sh` command like this: `./run_tests.sh --pdb`. This will drop you to the PDB shell.

3.1.1 Requirements

This script expects that package pytest is installed. In case you don't have it yet, it can be easily installed using following command:

```
pip install --user pytest
```

or for all users:

```
sudo pip install pytest
```

Note: Some of the tests also require file `login` placed in the `tests/` directory. If you don't provide this file, tests from `test_user.py` will fail.

Example of the `login` file:

```
username
password
```

3.1.2 Example

```
$ ./run_tests.sh
============================= test session starts ==============================
platform linux2 -- Python 2.7.6 -- py-1.4.27 -- pytest-2.7.0
rootdir: /home/Dropbox/c0d3z/python/interface/abclinuxuapi, inifile:
collected 33 items
```

```
tests/test_blogpost.py ..........
tests/test_comment.py ..........
tests/test_init.py ...
tests/test_shared.py ...
tests/test_user.py .......

========================== 33 passed in 16.07 seconds ==========================
```

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index