
PyInstapaper Documentation

Release 0.2.2

Matt Dorn

May 07, 2018

Contents:

1	PyInstapaper	1
1.1	Usage	1
1.2	Installation	2
1.3	Additional info	2
2	Installation	3
2.1	Stable release	3
2.2	From sources	3
3	Usage	5
4	pyinstapaper	7
4.1	pyinstapaper package	7
5	Contributing	11
5.1	Types of Contributions	11
5.2	Get Started!	12
5.3	Pull Request Guidelines	13
5.4	Tips	13
5.5	Deploying	13
6	Credits	15
6.1	Development Lead	15
6.2	Contributors	15
7	History	17
7.1	0.1.0 (2015-09-26)	17
7.2	0.2.0 (2018-05-02)	17
8	Indices and tables	19
	Python Module Index	21

CHAPTER 1

PyInstapaper

Instapaper is a tool for saving web pages to read later, e.g. offline on a mobile device. PyInstapaper is a Python wrapper for the [full Instapaper API](#).

To use it, in addition to your Instapaper account username and password, you'll need to request API credentials from Instapaper.

1.1 Usage

```
from pyinstapaper.instapaper import Instapaper, Folder

INSTAPAPER_KEY = 'MY_INSTAPAPER_API_KEY'
INSTAPAPER_SECRET = 'MY_INSTAPAPER_API_SECRET'
INSTAPAPER_LOGIN = 'me@example.com'
INSTAPAPER_PASSWORD = 'p@ssw0rd'

instapaper = Instapaper(INSTAPAPER_KEY, INSTAPAPER_SECRET)
instapaper.login(INSTAPAPER_LOGIN, INSTAPAPER_PASSWORD)

# Get the 10 latest instapaper bookmarks for the given account and do
# something with the article text
bookmarks = instapaper.get_bookmarks('starred')
for bookmark in enumerate(bookmarks):
    do_something(bookmark.get_text())
    bookmark.archive()

# Create a new folder
folder = Folder(instapaper, title='cool stuff')
result = folder.add()
```

1.2 Installation

To install PyInstapaper, simply:

```
pip install pyinstapaper
```

1.3 Additional info

- Free software: MIT License
- Documentation: <https://pyinstapaper.readthedocs.org>.
- Boilerplate courtesy of cookiecutter: <https://github.com/audreyr/cookiecutter>
- Thanks to [Ryan Galloway](#) for pointing the way to using the Python oauth2 library for Instapaper's XAuth implementation, though ideally pyinstapaper would use [requests-oauthlib](#)

2.1 Stable release

To install PyInstapaper, run this command in your terminal:

```
$ pip install pyinstapaper
```

This is the preferred method to install PyInstapaper, as it will always install the most recent stable release.

If you don't have [pip](#) installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for PyInstapaper can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/mdorn/pyinstapaper
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/mdorn/pyinstapaper/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```


CHAPTER 3

Usage

To use PyInstapaper in a project:

```
import pyinstapaper
```


4.1 pyinstapaper package

4.1.1 Submodules

4.1.2 pyinstapaper.instapaper module

```
class pyinstapaper.instapaper.Bookmark(client, **data)
    Bases: pyinstapaper.instapaper.InstapaperObject
    Object representing an Instapaper bookmark/article.

    ATTRIBUTES = ['bookmark_id', 'title', 'description', 'hash', 'url', 'progress_timestamp']
    RESOURCE = 'bookmarks'
    RESOURCE_ID_ATTRIBUTE = 'bookmark_id'
    SIMPLE_ACTIONS = ['delete', 'star', 'archive', 'unarchive', 'get_text']
    TIMESTAMP_ATTRS = ['progress_timestamp', 'time']

    get_highlights()
        Get highlights for Bookmark instance.

        Returns list of Highlight objects

        Return type list

class pyinstapaper.instapaper.Folder(client, **data)
    Bases: pyinstapaper.instapaper.InstapaperObject
    Object representing an Instapaper folder.

    ATTRIBUTES = ['folder_id', 'title', 'display_title', 'sync_to_mobile', 'folder_id', 'p
    RESOURCE = 'folders'
    RESOURCE_ID_ATTRIBUTE = 'folder_id'
```

```
SIMPLE_ACTIONS = ['delete']
```

```
set_order(folder_ids)
    Order the user's folders
```

Parameters **folders** (*list*) – List of folder IDs in the desired order.

Returns List Folder objects in the new order.

Return type list

```
class pyinstapaper.instapaper.Highlight(client, **data)
```

Bases: *pyinstapaper.instapaper.InstapaperObject*

Object representing an Instapaper highlight.

```
ATTRIBUTES = ['highlight_id', 'text', 'note', 'time', 'position', 'bookmark_id', 'type']
```

```
RESOURCE = 'highlights'
```

```
RESOURCE_ID_ATTRIBUTE = 'highlight_id'
```

```
SIMPLE_ACTIONS = ['delete']
```

```
TIMESTAMP_ATTRS = ['time']
```

```
create()
```

```
class pyinstapaper.instapaper.Instapaper(oauth_key, oauth_secret)
```

Bases: object

Instapaper client class.

Parameters

- **str** (*oauth_secret*) – Instapaper OAuth consumer key
- **str** – Instapaper OAuth consumer secret

```
get_bookmarks(folder='unread', limit=25, have=None)
```

Return list of user's bookmarks.

Parameters

- **folder** (*str*) – Optional. Possible values are unread (default), starred, archive, or a folder_id value.
- **limit** (*int*) – Optional. A number between 1 and 500, default 25.
- **have** (*list*) – Optional. A list of IDs to exclude from results

Returns List of user's bookmarks

Return type list

```
get_folders()
```

Return list of user's folders.

Return type list

```
login(username, password)
```

Authenticate using XAuth variant of OAuth.

Parameters

- **username** (*str*) – Username or email address for the relevant account
- **password** (*str*) – Password for the account

request (*path*, *params=None*, *returns_json=True*, *method='POST'*, *api_version='1'*)

Process a request using the OAuth client's request method.

Parameters

- **path** (*str*) – Path fragment to the API endpoint, e.g. “resource/ID”
- **params** (*dict*) – Parameters to pass to request
- **method** (*str*) – Optional HTTP method, normally POST for Instapaper
- **api_version** (*str*) – Optional alternative API version

Returns response headers and body

Retval dict

class pyinstapaper.instapaper.**InstapaperObject** (*client*, ***data*)

Bases: object

Base class for Instapaper objects like Bookmark.

Parameters

- **client** (*oauth2.Client*) – instance of the OAuth client for making requests
- **data** (*dict*) – key/value pairs of object attributes, e.g. title, etc.

add()

Save an object to Instapaper after instantiating it.

Example:

```
folder = Folder(instapaper, title='stuff')
result = folder.add()
```

4.1.3 Module contents

Top-level package for PyInstapaper.

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at <https://github.com/mdorn/pyinstapaper/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

5.1.4 Write Documentation

PyInstapaper could always use more documentation, whether as part of the official PyInstapaper docs, in docstrings, or even on the web in blog posts, articles, and such.

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/mdorn/pyinstapaper/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

5.2 Get Started!

Ready to contribute? Here's how to set up *pyinstapaper* for local development.

1. Fork the *pyinstapaper* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/pyinstapaper.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv pyinstapaper
$ cd pyinstapaper/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 pyinstapaper tests
$ python setup.py test or py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7, 3.4, 3.5 and 3.6, and for PyPy. Check https://travis-ci.org/mdorn/pyinstapaper/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ py.test tests.test_pyinstapaper
```

5.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

6.1 Development Lead

- Matt Dorn <mdorn@textmethod.com>

6.2 Contributors

None yet. Why not be the first?

7.1 0.1.0 (2015-09-26)

- First release on PyPI.

7.2 0.2.0 (2018-05-02)

- Python 3 compatibility
- Added 'have' param to get_bookmarks

CHAPTER 8

Indices and tables

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

p

`pyinstapaper`, 9

`pyinstapaper.instapaper`, 7

A

add() (pyinstapaper.instapaper.InstapaperObject method), 9

ATTRIBUTES (pyinstapaper.instapaper.Bookmark attribute), 7

ATTRIBUTES (pyinstapaper.instapaper.Folder attribute), 7

ATTRIBUTES (pyinstapaper.instapaper.Highlight attribute), 8

B

Bookmark (class in pyinstapaper.instapaper), 7

C

create() (pyinstapaper.instapaper.Highlight method), 8

F

Folder (class in pyinstapaper.instapaper), 7

G

get_bookmarks() (pyinstapaper.instapaper.Instapaper method), 8

get_folders() (pyinstapaper.instapaper.Instapaper method), 8

get_highlights() (pyinstapaper.instapaper.Bookmark method), 7

H

Highlight (class in pyinstapaper.instapaper), 8

I

Instapaper (class in pyinstapaper.instapaper), 8

InstapaperObject (class in pyinstapaper.instapaper), 9

L

login() (pyinstapaper.instapaper.Instapaper method), 8

P

pyinstapaper (module), 9

pyinstapaper.instapaper (module), 7

R

request() (pyinstapaper.instapaper.Instapaper method), 8

RESOURCE (pyinstapaper.instapaper.Bookmark attribute), 7

RESOURCE (pyinstapaper.instapaper.Folder attribute), 7

RESOURCE (pyinstapaper.instapaper.Highlight attribute), 8

RESOURCE_ID_ATTRIBUTE (pyinstapaper.instapaper.Bookmark attribute), 7

RESOURCE_ID_ATTRIBUTE (pyinstapaper.instapaper.Folder attribute), 7

RESOURCE_ID_ATTRIBUTE (pyinstapaper.instapaper.Highlight attribute), 8

S

set_order() (pyinstapaper.instapaper.Folder method), 8

SIMPLE_ACTIONS (pyinstapaper.instapaper.Bookmark attribute), 7

SIMPLE_ACTIONS (pyinstapaper.instapaper.Folder attribute), 7

SIMPLE_ACTIONS (pyinstapaper.instapaper.Highlight attribute), 8

T

TIMESTAMP_ATTRS (pyinstapaper.instapaper.Bookmark attribute), 7

TIMESTAMP_ATTRS (pyinstapaper.instapaper.Highlight attribute), 8