Pixiv Documentation

Release 0.1.1

Louis Taylor

Jun 11, 2017

Contents

CHAPTER 1

python-pixiv

python-pixiv: Pixiv API client for moe girls.

- Free software: LGPLv3
- Documentation: https://pixiv.readthedocs.org.
- Contribute: https://github.com/kragniz/python-pixiv

python-pixiv supports and runs continuous tests for python 2.7, 3.4 and 3.5, and PyPy. Tests are run on both Linux and Windows.

Quickstart

Install python-pixiv:

```
$ pip install pixiv
```

Login to pixiv:

```
from pixiv import login

pixiv = login('username', 'password')
```

Save the work from a particular user:

```
user = pixiv.user(7631951)

for art in user.works():
    art.save()
```

See the full documentation for more!

CHAPTER 2

Contents:

Installation

At the command line:

```
$ pip install pixiv
```

Usage

To use Pixiv in a project:

```
import pixiv
```

Example

Warning: This is for demonstration purposes only, and not currently functional.

```
from pixiv import login

pixiv = login('weeb', password='hunter2')
pixiv.me
pixiv.me.following

user = pixiv.user(171980)

for work in user.works:
    print(work.title)
```

```
user.favorites

for art in user.works():
    # save the artwork to the currect working directory
    art.save()
```

API reference

pixiv.**login**(*username*, *password*, *session=None*)

class pixiv.**Pixiv**(*session=None*)

Bases: pixiv.pixiv.Authed

Store session data

login(*username*, *password*)

Logs the user into Pixiv.

Parameters

- **username** (*str*) – login name
- **password** (*str*) – password for the login

search(*terms*, *period='all'*, *order='asc'*)

Search pixiv and return a list of *Work* objects.

Parameters

- **terms** (*str*) – search terms
- **period** (*str*) – period to search over. This must be one of `'all'`, `'day'`, `'week'` or `'month'`
- **order** (*str*) – sort order to list results. This must be either `'asc'` or `'desc'`

user(*user_id*)

Return a *User* object for a particular Pixiv user.

Parameters **user_id** (*int*) – ID of the user

Return type *User*

work(*work_id*)

Return a *Work* object with a specified ID.

Parameters **work_id** (*int*) – ID of the artwork

Return type *Work*

class pixiv.**User**(*id*, *auth_token=None*, *session=None*)

Bases: pixiv.pixiv.BaseUser, pixiv.pixiv.Authed

A Pixiv user

Parameters **id** (*int*) – the id of this user

works()

Return a list of *Work* created by this user

class `pixiv.`**`Work`**(*id*, *auth_token=None*, *session=None*)
Bases: `pixiv.pixiv.Authed`

A Pixiv artwork

Parameters **`id`** (*int*) – the id of this work

Variables

- **`id`** (*int*) – ID of this work
- **`image`** (*str*) – URL of the large size image for this work
- **`width`** (*int*) – width of image
- **`height`** (*int*) – height of image
- **`tags`** – list of tags this image has been tagged with

classmethod **`from_api_data`**(*api_data*, *auth_token=None*, *session=None*)
Return a new instance populated with data from the API

`link`

`save`(*filename=None*)
Save this artwork to a local file

Parameters **`filename`** (*str*) – the filename to save to. If this is `None`, then the image will be named with the default from the pixiv site, e.g. `1234567_p0.jpg`

Returns the filename the image was saved to

Return type str

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

Types of Contributions

Report Bugs

Report bugs at https://github.com/kragniz/python-pixiv/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.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with "feature" is open to whoever wants to implement it.

Write Documentation

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

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/kragniz/python-pixiv/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 :)

Get Started!

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

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

```
$ git clone git@github.com:your_name_here/python-pixiv.git
$ cd python-pixiv
```

3. Install your local copy into a virtualenv:

```
$ virtualenv env
$ source env/bin/activate
$ pip install -e .
```

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:

```
$ 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.

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, and 3.5, and for PyPy. Check https://travis-ci.org/kragniz/python-pixiv/pull_requests and make sure that the tests pass for all supported Python versions. We use six for compatibility in the parts where the python2 and python3 APIs diverge. Use this instead of rolling your own compatibility layer.

Tips

To run a subset of tests:

```
$ python -m unittest tests.test_pixiv
```

Credits

Development Lead

- Louis Taylor <louis@kragniz.eu>

Contributors

None yet. Why not be the first?

History

0.1.0 (2015-01-20)

- First release on PyPI.
- Basic things like logging in and viewing a list of works a user has created work.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

Index