
anapioficeandfire Documentation

Release 0.1.2

Joakim Skoog

November 15, 2016

1	Getting started	3
1.1	Introduction	3
1.2	Installation	3
1.3	Usage	3
1.4	First example	3
1.5	API	3
1.6	Models	4
2	API Reference	5
3	Models Reference	7
4	Cursor	9
5	Contributing	11
5.1	Types of Contributions	11
5.2	Get Started!	12
5.3	Pull Request Guidelines	12
5.4	Tips	13
6	Credits	15
6.1	Development Lead	15
6.2	Contributors	15
7	History	17
7.1	0.1.0 (2016-3-3)	17
7.2	0.1.1 (2016-3-3)	17
7.3	0.1.2 (2016-3-3)	17
8	Indices and tables	19

Contents:

Getting started

1.1 Introduction

This is the place for people that are new to `anapioficeandfire`. This tutorial will help you to get started with `anapioficeandfire`. Note that we won't go too much into detail, just the most important basic functionality.

1.2 Installation

At the command line:

```
$ pip install anapioficeandfire
```

1.3 Usage

To use `anapioficeandfire` in a project:

```
import anapioficeandfire
```

1.4 First example

```
import anapioficeandfire

api = anapioficeandfire.API()

jon_snow = api.get_character(id=583)
for title in jon_snow.aliases:
    print(title)
```

This example will download all data about the character Jon Snow and print each one of his aliases to the console.

1.5 API

The API class provides access to the entire An API of Ice And Fire in a clean and “pythonic” way. Each method accepts various parameters and return responses. For detailed information about the methods, please refer to *API*

Reference.

1.6 Models

When an API method is invoked the response will be an anapiofireandfire model class instance. The model will contain the data returned from An API of Ice And Fire which you can then use inside your applications. For example, the following code returns a `Book` model:

```
# Get a Book object form An API of Ice And Fire
game_of_thrones = api.get_book(id=1)
```

For detailed information about the models, please refer to *Models Reference*.

API Reference

This page contains detailed documentation for the `anapiofireandfire` module.

Models Reference

This page contains detailed documentation for all models in the anapiofireandfire module.

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:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at <https://github.com/joakimskoog/anapioficeandfire-python/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” is open to whoever wants to implement it.

5.1.3 Implement Features

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

5.1.4 Write Documentation

anapioficeandfire-python could always use more documentation, whether as part of the official anapioficeandfire-python 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/joakimskoog/anapioficeandfire-python/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 *anapioficeandfire-python* for local development.

1. Fork the *anapioficeandfire-python* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/anapioficeandfire-python.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 anapioficeandfire-python
$ cd anapioficeandfire-python/
$ 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 anapioficeandfire-python tests
$ python setup.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 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/joakimskoog/anapioficeandfire-python/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

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

Credits

6.1 Development Lead

- Joakim Skoog <joakimskooog@gmail.com>

6.2 Contributors

None yet. Why not be the first?

History

7.1 0.1.0 (2016-3-3)

- First release on PyPI.

7.2 0.1.1 (2016-3-3)

- Fixed a few things

7.3 0.1.2 (2016-3-3)

- Importing submodules in init

Indices and tables

- `genindex`
- `search`