
djcurrencies Documentation

Release 1.1.0

Lihan

Jan 16, 2023

Contents

| | | |
|-----------|-----------------------------------|-----------|
| 1 | Documentation | 3 |
| 2 | Quickstart | 5 |
| 2.1 | Settings | 5 |
| 3 | Features | 7 |
| 4 | Running Tests | 9 |
| 5 | Installation | 11 |
| 6 | Usage | 13 |
| 7 | Contributing | 15 |
| 7.1 | Types of Contributions | 15 |
| 7.2 | Get Started! | 16 |
| 7.3 | Pull Request Guidelines | 17 |
| 7.4 | Tips | 17 |
| 8 | History | 19 |
| 9 | 0.0.1 (2018-04-23) | 21 |
| 10 | 1.0.1 (2020-03-09) | 23 |
| 11 | 1.1.0 (2022-01-17) | 25 |

Contents:

CHAPTER 1

Documentation

The full documentation is at <https://dj-currencies.readthedocs.io>.

CHAPTER 2

Quickstart

For Django 2 support, please use version 0.1.2. Django 3 support added in version 1.0. Django 4 support from version >=1.1.

Install djcurrencies:

```
pip install dj-currencies
```

Add it to your *INSTALLED_APPS*:

```
INSTALLED_APPS = (  
    ...  
    'dj_currencies',  
    ...  
)
```

2.1 Settings

```
DJANGO_CURRENCIES = {  
    'DEFAULT_BACKEND': 'djmoney_rates.backends.OpenExchangeBackend',  
    'OPENEXCHANGE_APP_ID': '',  
    'BASE_CURRENCIES': ['USD'],  
    'MAX_CACHE_DAYS': 7  
}
```

DEFAULT_BACKEND: The selected backend to sync exchange rates

OPENEXCHANGE_APP_ID: Must be configured if you use **OpenExchangeBackend**

BASE_CURRENCIES: A list of base currencies to use. At the time of this version, you will only be able to convert currency from any one of the base currency to target currency.

MAX_CACHE_DAYS: Only use the cache within this time limit. If exchange rates was not synced within the time frame, an exception will thrown

Note: You will need to have at least “OPENEXCHANGE_APP_ID” configured if you use **OpenExchangeBackend**

CHAPTER 3

Features

- [open exchange rates](openexchangerates.org) integration
- Extensible backend design, hook your own exchange rate sources
- Multi base currencies support, no double conversion to lose precision
- Store historical exchange rates
- offline currency conversion

CHAPTER 4

Running Tests

Does the code actually work?

```
source <YOURVIRTUALENV>/bin/activate  
(myenv) $ python runtests.py
```


CHAPTER 5

Installation

At the command line:

```
$ pip install dj-currencies
```


CHAPTER 6

Usage

To use `djcurrencies` in a project, add it to your *INSTALLED_APPS*:

```
INSTALLED_APPS = (  
    ...  
    'dj_currencies',  
    ...  
)
```


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

7.1 Types of Contributions

7.1.1 Report Bugs

Report bugs at <https://github.com/CoverGenius/dj-currencies/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.

7.1.2 Fix Bugs

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

7.1.3 Implement Features

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

7.1.4 Write Documentation

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

7.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/lihan/dj-currencies/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 :)

7.2 Get Started!

Ready to contribute? Here's how to set up *dj-currencies* for local development.

1. Fork the *dj-currencies* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/dj-currencies.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 dj-currencies
$ cd dj-currencies/
$ 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 dj_currencies tests
$ python runtests.py
$ 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.

7.3 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 3.8+. Ensure builds are passing on GitHub.

7.4 Tips

To run a subset of tests:

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


CHAPTER 8

History

CHAPTER 9

0.0.1 (2018-04-23)

- First release on PyPI.

CHAPTER 10

1.0.1 (2020-03-09)

- Added Django 3.1 Support.

CHAPTER 11

1.1.0 (2022-01-17)

- Django 4.0+ support.