
django-money-rates Documentation

Release 0.3.1

Federico Frenguelli

Jun 27, 2017

Contents

1	django-money-rates	3
1.1	Documentation	3
1.2	Quickstart	3
1.3	Setup the Open Exchange Rates backend	4
1.4	Pull the latest Exchange Rates	4
1.5	Convert from one currency to another	4
1.6	Features	4
1.7	TODO List	4
2	Installation	5
3	Usage	7
4	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	11
4.4	Tips	11
5	Credits	13
5.1	Development Lead	13
5.2	Contributors	13
6	History	15
6.1	0.3.0 (2013-12-30)	15
6.2	0.1.0 (2013-10-16)	15

Contents:

Currency conversion for money

Documentation

The full documentation is at <https://django-money-rates.readthedocs.io/>.

Quickstart

Install django-money-rates:

```
pip install django-money-rates
```

Then use it in a project:

```
import djmoney_rates
```

In order to save exchange rates to your database, add *djmoney_rates* to your `INSTALLED_APPS` in your project's settings:

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

Setup the Open Exchange Rates backend

Open an account at <https://openexchangerates.org/> if you don't have one already. Then, add this to your project's settings:

```
DJANGO_MONEY_RATES = {
    'DEFAULT_BACKEND': 'djmoney_rates.backends.OpenExchangeBackend',
    'OPENEXCHANGE_URL': 'http://openexchangerates.org/api/latest.json',
    'OPENEXCHANGE_APP_ID': 'YOUR APP ID HERE',
    'OPENEXCHANGE_BASE_CURRENCY': 'USD',
}
```

For more information on the Open Exchange Rates API, see <https://openexchangerates.org/>

Pull the latest Exchange Rates

Once your backend is setup, get the latest exchange rates:

```
$ ./manage.py update_rates
```

Convert from one currency to another

Here's an example of converting 10 Euros to Brazilian Reais:

```
from moneyed import Money
from djmoney_rates.utils import convert_money
brl_money = convert_money(10, "EUR", "BRL")
```

Features

- Convert money from one currency to another with an easy to use API.

TODO List

- Add money converter wrapper for util's *convert_money* function.
- Add celery periodic task for getting daily exchange rates.

CHAPTER 2

Installation

At the command line:

```
$ easy_install django-money-rates
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv django-money-rates  
$ pip install django-money-rates
```


CHAPTER 3

Usage

To use django-money-rates in a project:

```
import djmoney_rates
```

TODO: add usages examples and step-by-step creation of a custom backend

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/evonove/django-money-rates/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

django-money-rates could always use more documentation, whether as part of the official django-money-rates 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/evonove/django-money-rates/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 *django-money-rates* for local development.

1. Fork the *django-money-rates* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/django-money-rates.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 django-money-rates
$ cd django-money-rates/
$ 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 djmoney_rates 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.

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, and 3.3, and for PyPy. Check https://travis-ci.org/evonove/django-money-rates/pull_requests and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

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


Development Lead

- Federico Frenguelli <synasius@gmail.com>

Contributors

- Tim Zenderman <tzenderman@gmail.com>
- Federico Dolce <psykopear@gmail.com>
- Adam Chainz <me@adamj.eu>

0.3.0 (2013-12-30)

- *convert_money* utility function now returns `moneyed.Money` instances

0.1.0 (2013-10-16)

- First release on PyPI.