Django plans payments Documentation

Release 1.1.1

Petr Dlouhý

Mar 29, 2023

Contents

1 Django plans payments **3**
- 1.1 Documentation . . . 3
- 1.2 Quickstart . . . 3
- 1.3 Customer IP address . . . 4
- 1.4 Running Tests . . . 4
- 1.5 Credits . . . 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 1.1.2 (2023-03-29) . . . 15
- 6.2 1.1.1 (2023-01-27) . . . 15
- 6.3 1.1.0 (2023-01-27) . . . 15
- 6.4 1.0.1 (2022-12-09) . . . 15
- 6.5 1.0.0 (2022-12-08) . . . 15
- 6.6 0.2.0 (2018-08-05) . . . 15
- 6.7 0.1.0 (2018-07-23) . . . 16

Contents:

CHAPTER 1

Django plans payments

Almost automatic integration between django-plans and django-payments. This will add payment buttons to the order page and automatically confirm the *Order* after the payment.

1.1 Documentation

The full documentation is at https://django-plans-payments.readthedocs.io.

1.2 Quickstart

Install and configure `django-plans` and `django-payments` apps. Capture mode is not yet supported, so `PAYMENT_VARINANTS` with `'capture': False` will not get confirmed.

Install Django plans payments:

```
pip install django-plans-payments
```

Add it to your `INSTALLED_APPS`, before the `plans`:

```
INSTALLED_APPS = (
    ...
    'related_admin',
    'plans_payments',
    'plans',
    ...
)
```

Add Django `plans_payments` to the URL patterns:

```
urlpatterns = [
    ...
    url(r'^plans-payments', include('plans_payments.urls')),
    ...
]
```

Set `django-plans` settings and set model to:

```
PAYMENT_MODEL = 'plans_payments.Payment'
```

1.3 Customer IP address

Customer IP address is stored in Payment model and used for some payment providers (i.e. PayU). For security reasons *django-plans-payments* does acquire the IP only from `request REMOTE_ADDR` parameter. If you are behind proxy, you will need to setup some mechanism to populate this variable from `HTTP_X_FORWARDED_FOR` parameter. The suggested solution is to use django-httpforwardedfor or django-xff application for that.

1.4 Running Tests

Does the code actually work?

```
source <YOURVIRTUALENV>/bin/activate
(myenv) $ pip install tox
(myenv) $ tox
```

1.5 Credits

Tools used in rendering this package:

- Cookiecutter
- cookiecutter-djangopackage

CHAPTER 2

Installation

At the command line:

```
$ easy_install django-plans-payments
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv django-plans-payments
$ pip install django-plans-payments
```

CHAPTER 3

Usage

To use Django plans payments in a project, add it to your *INSTALLED_APPS*:

```
INSTALLED_APPS = (
    ...
    'plans_payments.apps.PlansPaymentsConfig',
    ...
)
```

Add Django plans payments's URL patterns:

```
from plans_payments import urls as plans_payments_urls


urlpatterns = [
    ...
    url(r'^', include(plans_payments_urls)),
    ...
]
```

CHAPTER 4

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:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/PetrDlouhy/django-plans-payments/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.

4.1.2 Fix Bugs

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

4.1.3 Implement Features

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

4.1.4 Write Documentation

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

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/PetrDlouhy/django-plans-payments/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 :)

4.2 Get Started!

Ready to contribute? Here's how to set up *django-plans-payments* for local development.

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

```
$ git clone git@github.com:your_name_here/django-plans-payments.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-plans-payments
$ cd django-plans-payments/
$ 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 plans_payments 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.

4.3 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, and 3.3, and for PyPy. Check https://travis-ci.org/PetrDlouhy/django-plans-payments/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

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

CHAPTER 5

Credits

5.1 Development Lead

- Petr Dlouhý <petr.dlouhy@email.cz>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

History

6.1 1.1.2 (2023-03-29)

- reword Payment description to ommit word “Subscribtion” which might raise warnings for banks/card providers

6.2 1.1.1 (2023-01-27)

- correction release, include wheel update, correctly rebase to master

6.3 1.1.0 (2023-01-27)

- Fix transaction fee double counting

6.4 1.0.1 (2022-12-09)

- Fix migrations

6.5 1.0.0 (2022-12-08)

- Recurring payments functionality

6.6 0.2.0 (2018-08-05)

- Payment process without capturing should work

- Automatic buttons generation

6.7 0.1.0 (2018-07-23)

- First release on PyPI.