
django-getpaid-epay Documentation

Release 0.1.4

German Ilyin

May 25, 2016

1	django-getpaid-epay	3
1.1	Documentation	3
1.2	Features	3
1.3	Dependencies	3
1.4	Quickstart	4
1.5	Basic configuration	4
1.6	Credits	4
2	Installation	5
3	Usage	7
3.1	Configuration	7
3.2	Keys	8
3.3	Testing callback	8
4	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	10
4.4	Tips	11
5	Credits	13
5.1	Development Lead	13
5.2	Contributors	13
6	History	15
6.1	0.1.0 (2016-02-22)	15
6.2	0.1.3 (2016-04-22)	15
6.3	0.1.3 (2016-04-22)	15

Contents:

django-getpaid-epay



A django-getpaid payment backend for Kazkommertsbank Epay

1.1 Documentation

The full documentation is at <https://django-getpaid-epay.readthedocs.org>.

1.2 Features

- Authenticate payment
- Capture blocked money from admin
- Reverse payment from admin
- Refund payment from admin
- Check status from admin

1.3 Dependencies

Here is a list of dependencies:

```
# obvious
django
django-getpaid

M2Crypto
django-model-utils
xmltodict
requests
```

Warning: You might need to install some system packages to compile M2Crypto. What to install?

1.4 Quickstart

Install django-getpaid and then install django-getpaid-epay:

```
pip install django-getpaid-epay
```

Add `epay` to `INSTALLED_APPS` and `GETPAID_BACKENDS` in your django project. Do not forget to configure the `GETPAID_ORDER_MODEL` setting.

Run migration:

```
./manage.py migrate
```

1.5 Basic configuration

Here is minimal configuration:

```
GETPAID_BACKENDS_SETTINGS = {
    'epay' : {
        "key_passphrase" : 'your private key passphrase', # can be omitted if you have removed passphrase
        "merchant_private_key" : 'path to your private key',
        "merchant_id" : "92061101",
        "merchant_name" : "Test Shop",
        "merchant_cert_id" : "00c182b189"
        # "kkb_pub_key": 'path to Kazkom Public Key (kkbca.pem)', # optional
        # "modify_order_id": lambda id: id + 100000,
        # "unmodify_order_id": lambda id: id - 100000,
        # 'scheme': 'http',
        # 'testing' : True, # optional
    },
}
```

1.6 Credits

Tools used in rendering this package:

- Cookiecutter
- cookiecutter-djangopackage

Installation

At the command line:

```
$ easy_install django-getpaid-epay
```

Or, if you have pip:

```
$ pip install django-getpaid-epay
```

Usage

Note: I assume that you already have django-getpaid installed and configured

Install django-getpaid-epay:

```
pip install django-getpaid-epay
```

Add epay to `INSTALLED_APPS` and `GETPAID_BACKENDS` in your django project. Do not forget to configure the `GETPAID_ORDER_MODEL` setting.

Run migration:

```
./manage.py migrate
```

3.1 Configuration

Note: Putting your private keys in codebase is bad practice. Keep them outside of your project folder.

Here is minimal configuration:

```
GETPAID_BACKENDS_SETTINGS = {
    'epay' : {
        "key_passphrase" : 'your private key passphrase', # optional
        "merchant_private_key" : 'path to your private key',
        "merchant_id" : "92061101",
        "merchant_name" : "Test Shop",
        "merchant_cert_id" : "00c182b189",
        "kkb_pub_key": 'path to Kazkom Public Key (kkbca.pem)', # optional
        "modify_order_id": lambda id: id + 100000, # optional
        "unmodify_order_id": lambda id: id - 100000, # optional
        'scheme': 'http', # optional
        'testing' : True, # optional
        'template': 'some_template', # optional
    },
}
```

Explanation:

- `key_passphrase` - Your private key passphrase. This can be omitted if you have removed passphrase, see below
- `merchant_private_key` - path to private key in file system.
- `merchant_id` - your merchant id
- `merchant_name` - your merchant name
- `merchant_cert_id` - your certificate id
- `kbb_pub_key` - path to kbb public key in file system
- `modify_order_id` - a function to modify order id sent to epay
- `unmodify_order_id` - the reverse of `modify_order_id`. Used when order id is receive from epay.
- `scheme` - Use http or https protocol for callback and fallback urls
- `testing` - boolean. use testing or production epay server
- `auto_capture` - boolean. Automatically capture blocked money
- `template` - template of epay payment page
- `get_email` - function. takes two arguments (request, payment). This value can be specified to override default logic.
- `get_language` - function. takes two arguments (request, payment). This value can be specified to override default logic.
- `get_template` - function. takes two arguments (request, payment). This value can be specified to override default logic.
- `get_fail_backlink` - function. takes two arguments (request, payment). This value can be specified to override default logic.
- `get_backlink_url` - function. takes two arguments (request, payment). This value can be specified to override default logic.

Note: `order_id` sent to epay should be unique. You cannot authenticate two different payments with same id. Sometimes it's useful to be able to modify `order_id` sent to epay, you can do it with `modify_order_id` and `unmodify_order_id` settings.

3.2 Keys

To remove passphrase from private key:

```
$ openssl rsa -in cert.prv -out cert.pem
```

3.3 Testing callback

To test Epay payment backend in real conditions you need a public address so that bank processing can request your callback url. You can use `ngrok` to get temporary public address during development or testing:

```
$ ngrok http 8000
$ GETPAID_SITE_DOMAIN=1d258d87.ngrok.io ./manage.py runserver
```

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/yunmanger1/django-getpaid-epay/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-getpaid-epay could always use more documentation, whether as part of the official django-getpaid-epay 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/yunmanger1/django-getpaid-epay/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-getpaid-epay* for local development.

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

```
$ git clone git@github.com:your_name_here/django-getpaid-epay.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-getpaid-epay
$ cd django-getpaid-epay/
$ 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 epay 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/yunmanger1/django-getpaid-epay/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_epay
```

Credits

5.1 Development Lead

- German Ilyin <germanilyin@gmail.com>

5.2 Contributors

None yet. Why not be the first?

History

6.1 0.1.0 (2016-02-22)

- First release on PyPI.

6.2 0.1.3 (2016-04-22)

- Added `template` setting
- Added `get_email` setting
- Added `get_language` setting
- Added `get_template` setting
- Added `get_fail_backlink` setting
- Added `get_backlink_url` setting
- Added `auto_capture` setting
- Updated test gateway url to <https://testpay.kkb.kz>

6.3 0.1.3 (2016-04-22)

- Bugfix *scheme* setting handling