
dj-paginator Documentation

Release 0.2.2

Daniel Roy Greenfeld

January 11, 2017

1	dj-paginator	3
1.1	Documentation	3
1.2	Features	3
1.3	Quickstart	3
1.4	Switching to a new theme method 1	4
1.5	Switching to a new theme method 2	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	10
4.4	Tips	11
5	Credits	13
5.1	Development Lead	13
5.2	Contributors	13
6	History	15
6.1	0.2.2 ???	15
6.2	0.2.2 (2015-10-18)	15
6.3	0.2.1 (2015-09-04)	15
6.4	0.1.0 (2015-07-26)	15

Contents:

A super-simple set of pagination tools for Django 1.8+

1.1 Documentation

The full documentation will soon be at <https://dj-paginator.readthedocs.io>.

1.2 Features

- Super simple code base.
- Easy to create and switch themes.
- Really good documentation! (coming soon)
- Really good tests! (coming soon)
- Class-Based View mixin so you don't need to load template tags (coming soon)

1.3 Quickstart

Install dj-paginator:

```
pip install dj-paginator
```

In your installed apps:

```
INSTALLED_APPS = [  
    'paginator'  
]
```

In your list view:

```
class MyListView(ListView):  
    model = MyModel  
    paginate_by = 10
```

In your list view template:

```
{% load paginator_tags %}

{% paginator %}
```

Done!

1.4 Switching to a new theme method 1

```
# Currently defaults to bootstrap.
# I'll add foundation soon.
PAGINATOR_THEME = 'foundation'
```

1.5 Switching to a new theme method 2

```
{% load paginator_tags %}

{# Is the default #}
{% bootstrap_paginator %}

{# For when I add the foundation theme#}
{% foundation_paginator %}
```

Installation

At the command line:

```
$ easy_install dj-paginator
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv dj-paginator  
$ pip install dj-paginator
```

Usage

To use dj-paginator in a project:

```
import paginator
```

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/pydanny/dj-paginator/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

dj-paginator could always use more documentation, whether as part of the official dj-paginator 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/pydanny/dj-paginator/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 *dj-paginator* for local development.

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

```
$ git clone git@github.com:your_name_here/dj-paginator.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-paginator
$ cd dj-paginator/
$ 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 paginator 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 2.7 and 3.3.. Check https://travis-ci.org/pydanny/dj-paginator/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_paginator
```

Credits

5.1 Development Lead

- Daniel Roy Greenfeld <pydanny@gmail.com>

5.2 Contributors

None yet. Why not be the first?

History

6.1 0.2.2 ???

- Attempted to create simple paginator
- Readme update

6.2 0.2.2 (2015-10-18)

- Use ellipsis for divider

6.3 0.2.1 (2015-09-04)

- Doc fix

6.4 0.1.0 (2015-07-26)

- First release on PyPI.
- Inception