

---

# **dj-paginator Documentation**

*Release 0.2.1*

**Daniel Roy Greenfeld**

**Sep 27, 2017**



---

# Contents

---

<b>1</b>	<b>dj-paginator</b>	<b>3</b>
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
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>Contributing</b>	<b>9</b>
4.1	Types of Contributions . . . . .	9
4.2	Get Started! . . . . .	10
4.3	Pull Request Guidelines . . . . .	11
4.4	Tips . . . . .	11
<b>5</b>	<b>Credits</b>	<b>13</b>
5.1	Development Lead . . . . .	13
5.2	Contributors . . . . .	13
<b>6</b>	<b>History</b>	<b>15</b>
6.1	0.1.0 (2015-07-26) . . . . .	15



Contents:



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

## Documentation

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

## 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)

## 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!

## Switching to a new theme method 1

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

## Switching to a new theme method 2

```
{% load paginator_tags %}

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

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



## CHAPTER 2

---

### Installation

---

At the command line:

```
$ easy_install dj-paginator
```

Or, if you have virtualenvwrapper installed:

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



## CHAPTER 3

---

### Usage

---

To use dj-paginator in a project:

```
import paginator
```



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/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.

#### 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

`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.

## 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 :)

## 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.

## 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](https://travis-ci.org/pydanny/dj-paginator/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_paginator
```





## Development Lead

- Daniel Roy Greenfeld <pydanny@gmail.com>

## Contributors

None yet. Why not be the first?



### **0.1.0 (2015-07-26)**

- First release on PyPI.
- Inception