
django-lineup Documentation

Release 0.3.1

abidibo

Dec 22, 2022

CONTENTS

1	Django Lineup	3
1.1	Documentation	3
1.2	Quickstart	3
1.3	Features	5
1.4	Running Tests	6
1.5	Development commands	6
1.6	Example app	6
1.7	Credits	6
2	Installation	7
3	Usage	9
4	Templatetags	11
4.1	Render Lineup Menu	11
4.2	Render Lineup Breadcrumbs	11
5	Management Commands	13
6	Contributing	15
6.1	Types of Contributions	15
6.2	Get Started!	16
6.3	Pull Request Guidelines	17
6.4	Tips	17
7	Credits	19
7.1	Development Lead	19
7.2	Contributors	19
8	History	21
8.1	0.3.1 (2022-12-22)	21
8.2	0.3.0 (2021-04-27)	21
8.3	0.2.3 (2021-02-28)	21
8.4	0.2.2 (2021-02-28)	21
8.5	0.2.1 (2020-12-14)	21
8.6	0.2.0 (2020-12-14)	21
8.7	0.1.0 (2020-12-11)	22

Contents:

DJANGO LINEUP

Multiple navigation system for django sites.

Django Lineup lets you manage a tree of items. Each first level node represents a menu you can include in your templates.

☐ Slug	Label	Parent	Link	Order	Login required	Enabled
☐ main-menu	Main Menu	-	-	0		
☐ home	Home	Main Menu	/home/	0		
☐ tools	Tools	Main Menu	-	1		
☐ dns-lookup	DNS Lookup	Tools	https://mxtoolbox.com/DNSLookup.aspx	0		
☐ dmarc	Dmarc	DNS Lookup	/dmarc/	0		
☐ google	Google	Tools	https://www.google.com	1		
☐ password-generator	Password Generator	Tools	https://www.lastpass.com/it/password-generator	2		
☐ draft	Draft	Main Menu	-	2		
☐ sidebar-menu	Sidebar Menu	-	-	1		
☐ news	News	Sidebar Menu	/news/	0		
☐ users	Users	Sidebar Menu	/admin/auth/user/	1		

1.1 Documentation

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

1.2 Quickstart

Install Django Lineup:

```
pip install django-lineup
```

Add it to your *INSTALLED_APPS*:

```
INSTALLED_APPS = (  
    ...  
    'lineup.apps.LineupConfig',  
    ...  
)
```

(continues on next page)

(continued from previous page)

```
)  
    ...  
)
```

Make sure the `requests` context processor is included (it is by default):

```
TEMPLATES = [  
    {  
        'OPTIONS': {  
            'context_processors': [  
                "django.template.context_processors.request",  
            ],  
        },  
    },  
]
```

Render a menu::

```
{% load lineup_tags %}  
{% lineup_menu 'my-root-item-slug' %}
```

Render the breadcrumbs::

```
{% load lineup_tags %}  
{% lineup_breadcrumbs 'my-root-item-slug' %}
```

Import a menu from a json::

```
$ python manage.py import_menu_from_json
```

Json example::

```
{  
  "label": "Main Menu",  
  "slug": "main-menu",  
  "order": 0,  
  "children": [  
    {  
      "label": "Tools",  
      "slug": "tools",  
      "order": 0,  
      "children": [  
        {  
          "label": "DNS Tools",  
          "slug": "dns-tools",  
          "order": 0,  
          "login_required": true,  
          "children": [  
            {  
              "label": "DMARC DNS Tools",  
              "slug": "dmarc-dns-tools",  
              "link": "/dmarc-tools/",  
              "title": "DMARC Rulez",  
              "order": 0  
            }  
          ]  
        }  
      ]  
    }  
  ]  
}
```

(continues on next page)

(continued from previous page)

```

        }
    ]
},
{
    "label": "Password Generator",
    "slug": "password-generator",
    "order": 1
}
]
},
{
    "label": "Disabled Item",
    "slug": "disabled-item",
    "order": 1,
    "enabled": false,
    "children": [
        {
            "label": "Disabled child",
            "slug": "disabled-child",
            "order": 0
        }
    ]
},
{
    "label": "Perm Item",
    "slug": "perm-item",
    "order": 2,
    "permissions": [
        "add_permission",
        "view_session"
    ]
}
]
}

```

1.3 Features

- Multiple menus supported
- Visibility logic: login required / permissions
- Render menu tree templatetags
- Breadcrumbs templatetag
- Import a menu from json management command
- [Django Baton](#) integration to highlight different menu in the admin

1.4 Running Tests

Does the code actually work?

```
source <YOURVIRTUALENV>/bin/activate
(myenv) $ pip install -r requirements_test.txt
(myenv) $ python runtests.py
```

1.5 Development commands

```
pip install -r requirements_dev.txt
invoke -l
```

1.6 Example app

This example is provided as a convenience feature to allow potential users to try the app straight from the app repo without having to create a django project.

It can also be used to develop the app in place.

To run this example, follow these instructions:

1. Navigate to the root directory of your application (same as *manage.py*)
2. Install the requirements for the package:

```
pip install -r requirements_test.txt
```
3. Make and apply migrations

```
python manage.py makemigrations
python manage.py migrate
```
4. Run the server

```
python manage.py runserver
```
5. Access from the browser at *http://127.0.0.1:8000*
6. Admin user account is admin:admin

1.7 Credits

Django Lineup is developed by Otto SRL.

Tools used in rendering this package:

- [Cookiecutter](#)
- [cookiecutter-djangopackage](#)

INSTALLATION

Install with pip:

```
$ pip install django-lineup
```


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

```
INSTALLED_APPS = (  
    ...  
    'lineup.apps.LineupConfig',  
    ...  
)
```

Be sure the `requests` context processor is included (it is by default):

```
TEMPLATES = [  
    {  
        'OPTIONS': {  
            'context_processors': [  
                # ...  
                "django.template.context_processors.request",  
            ],  
        },  
    },  
]
```


TEMPLATETAGS

Django Lineup provides 2 templatetags to include the rendered menu and the breadcrumbs in your templates.

4.1 Render Lineup Menu

Templatetag used to render a menu inside a template

```
{% load lineup_tags %}
{% lineup_menu 'main-menu' %}
```

It takes one parameter: the slug of the menu root voice.

4.2 Render Lineup Breadcrumbs

Templatetag used to render the breadcrumbs inside a template

```
{% load lineup_tags %}
{% lineup_breadcrumbs 'main-menu' %}
```

It takes one parameter: the slug of the menu root voice.

MANAGEMENT COMMANDS

Django Lineup provides a management command you can use to import a JSON formatted menu from file

```
$ python manage.py import_menu_from_json /path/to/menu.json
```

The file should contain a JSON representation of the menu, i.e.

```
{
  "label": "Main Menu",
  "slug": "main-menu",
  "order": 0,
  "children": [
    {
      "label": "Tools",
      "slug": "tools",
      "order": 0,
      "children": [
        {
          "label": "DNS Tools",
          "slug": "dns-tools",
          "order": 0,
          "login_required": true,
          "children": [
            {
              "label": "DMARC DNS Tools",
              "slug": "dmarc-dns-tools",
              "link": "/dmarc-tools/",
              "title": "DMARC Rulez",
              "order": 0
            }
          ]
        }
      ]
    },
    {
      "label": "Password Generator",
      "slug": "password-generator",
      "order": 1
    }
  ]
},
{
  "label": "Disabled Item",
  "slug": "disabled-item",
```

(continues on next page)

(continued from previous page)

```
"order": 1,
"enabled": false,
"children": [
    {
        "label": "Disabled child",
        "slug": "disabled-child",
        "order": 0
    }
]
},
{
    "label": "Perm Item",
    "slug": "perm-item",
    "order": 2,
    "permissions": [
        "add_permission",
        "view_session"
    ]
}
]
}
```

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:

6.1 Types of Contributions

6.1.1 Report Bugs

Report bugs at <https://github.com/otto-torino/django-lineup/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.

6.1.2 Fix Bugs

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

6.1.3 Implement Features

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

6.1.4 Write Documentation

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

6.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/otto-torino/django-lineup/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 :)

6.2 Get Started!

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

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

```
$ git clone git@github.com:your_name_here/django-lineup.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-lineup
$ cd django-lineup/
$ 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 lineup 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.

6.3 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/otto-torino/django-lineup/pull_requests and make sure that the tests pass for all supported Python versions.

6.4 Tips

To run a subset of tests:

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


CREDITS

7.1 Development Lead

- abidibo <abidibo@gmail.com>

7.2 Contributors

None yet. Why not be the first?

HISTORY

8.1 0.3.1 (2022-12-22)

- Adds Django 4 support

8.2 0.3.0 (2021-04-27)

- Adds the extras field

8.3 0.2.3 (2021-02-28)

- Fixes tests

8.4 0.2.2 (2021-02-28)

- Fixes baton row attributes method

8.5 0.2.1 (2020-12-14)

- Adds order field as editable in change list page
- First stable release

8.6 0.2.0 (2020-12-14)

- First stable release

8.7 0.1.0 (2020-12-11)

- First release on PyPI.