
django-djedit Documentation

Release 0.3.4

Roman

Mar 23, 2019

Contents

1	django-djedit	3
1.1	Screenshots	4
1.2	Supported Django versions	4
1.3	Quickstart	4
1.4	Credits	5
2	Installation	7
3	Configuration	9
3.1	Additional Options	10
3.2	SiteMap	10
4	Contributing	11
4.1	Types of Contributions	11
4.2	Get Started!	12
4.3	Pull Request Guidelines	13
4.4	Tips	13
5	Credits	15
5.1	Development Lead	15
5.2	Contributors	15
6	History	17
6.1	0.1.0 (2017-04-15)	17
6.2	0.1.5 (2017-05-26)	17
6.3	0.2 (2017-07-04)	17
6.4	0.3 (2018-06-02)	18

Contents:

CHAPTER 1

django-djreddit

Bare bones Django forum application with Reddit like features developed as a Django reusable app.

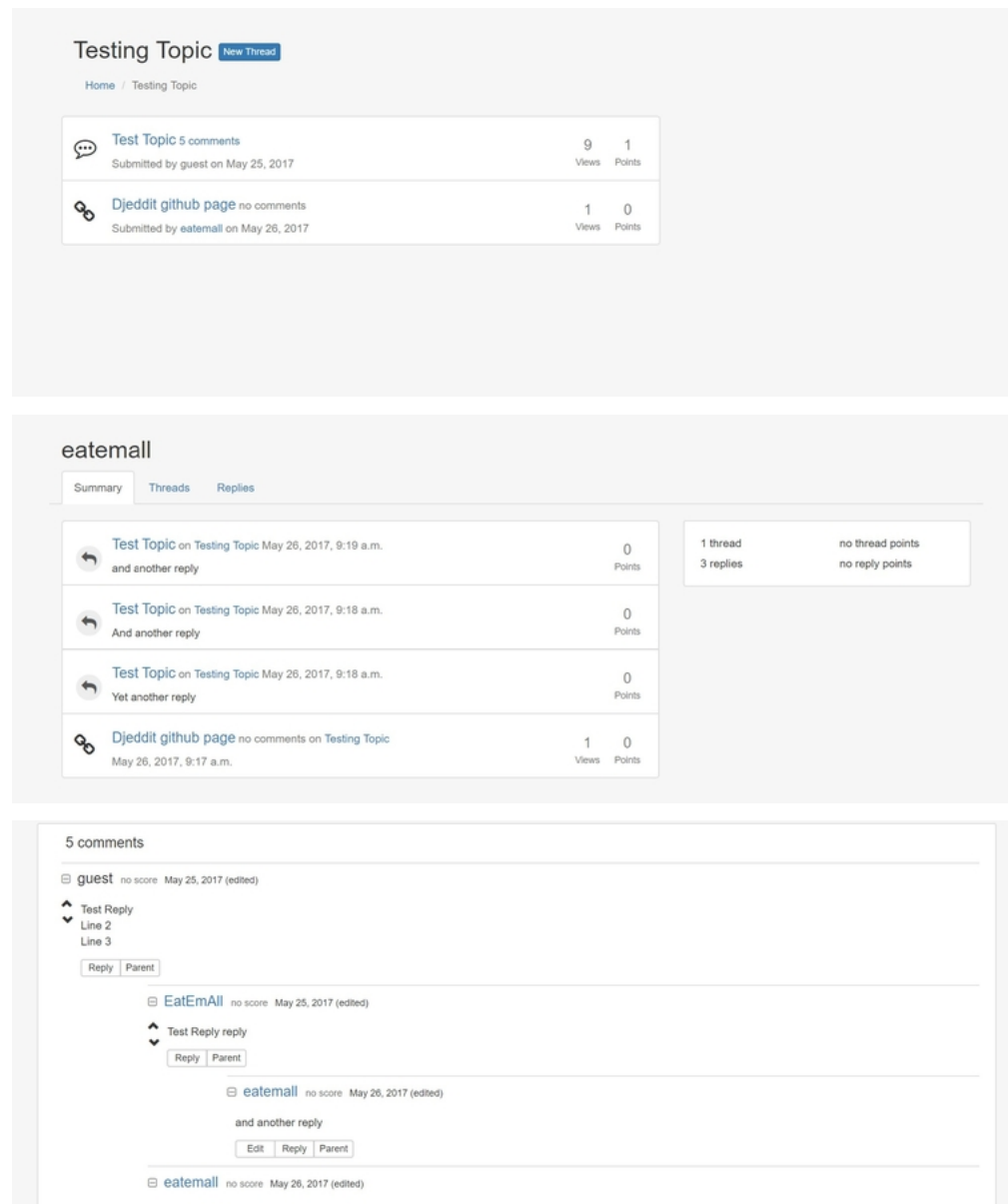
- comments are ranked using wilson scoring interval and displayed in a collapsable tree structure (using [django-mptt](#))
- voting functionality for threads & comments
- compatible with mobile screen sizes (using [Bootstrap](#))
- users management page for admins
- admins can lock/unlock, edit, delete threads and posts, edit & delete topics

[django-registration-redux](#) is recommended to go along with this app if you need out of the box user registration functionality.

Working demo: <http://eatemall.pythonanywhere.com>

Documentation: <https://django-djreddit.readthedocs.io>.

1.1 Screenshots



1.2 Supported Django versions

The app is currently tested with Django 1.8, 1.9, 1.10 and 1.11, 2.0

1.3 Quickstart

Install django-djedditi:


```
pip install django-djeddit
```

Add it and its dependencies to your *INSTALLED_APPS*:

```
INSTALLED_APPS = [
    ...
    'crispy_forms',
    'mptt',
    'djeddit',
    'meta',
    ...
]
```

djeddit_settings to context_processors:

```
'context_processors': [
    ...
    'djeddit.context_processors.djeddit_settings',
    ...
]
```

jango-djeddit's URL patterns:

```
urlpatterns = [
    ...
    url(r'^$', include('djeddit.urls')),
    ...
]
```

Migrate models:

```
python manage.py migrate djeddit
```

Create a topic:

You can use New Topic dialog in topics page if you're logged in as a superuser or you can create one in a python console:

```
from djeddit.models import Topic
Topic.objects.create(title='Test Topic')
```

1.4 Credits

Dependencies:

- django-mptt
- crispy_forms
- django-meta
- django-ipware

Tools used in rendering this package:

- Cookiecutter
- cookiecutter-djangopackage

CHAPTER 2

Installation

At the command line:

```
$ easy_install django-djedit
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv django-djedit  
$ pip install django-djedit
```


CHAPTER 3

Configuration

To use django-djedit in a project, add it to your *INSTALLED_APPS*:

```
INSTALLED_APPS = [
    ...
    'crispy_forms',
    'mptt',
    'djedit',
    'meta',
    ...
]
```

And djedit_settings to context_processors:

```
'context_processors': [
    ...
    'djedit.context_processors.djedit_settings',
    ...
]
```

Add django-djedit's URL patterns:

```
urlpatterns = [
    ...
    url(r'^$', include('djedit.urls')),
    ...
]
```

Migrate models:

```
python manage.py migrate djedit
```

3.1 Additional Options

By default the base template is djedit/base.html

You can change the default base template in settings.py:

```
DJEDIT_BASE_TEMPLATE = "path/to/template.html"
```

In order for the base template to work properly with the app's templates it needs to contain the following:

within <head>...</head>:

```
{% load staticfiles %}
{% include 'djedit/base_stylesheets.html' %}
```

within <body>...</body>:

```
{% block title %}{% endblock %}
{% block content %}{% endblock %}
{% block scripts %}{% endblock %}
{% include 'djedit/base_scripts.html' %}
{% block scripts %}{% endblock %}
```

You can use the structure of djedit/base.html for reference.

3.2 SiteMap

If you'd like djedit to generate sitemaps for SEO you can follow these steps. Djedit comes with a sitemaps.py file included and you just have to enable it.

Add the following apps to your installed apps if they are not already there

```
'django.contrib.sites',
'django.contrib.sitemaps',
```

Create the sitemaps dictionary with the djedit sitemap and core django sitemap imports in the urls.py of your project

Now add the following to your *urlpatterns*

```
url(r'^sitemap\.xml$', sitemap, {'sitemaps': sitemaps}, name='django.contrib.sitemaps.
↳ views.sitemap'),
```

Run migrations and run server

If you visit *sitemap.xml* on your site you should have a working sitemap for djedit threads.

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

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

```
$ git clone git@github.com:your_name_here/django-djeddit.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-djeddit
$ cd django-djeddit/
$ 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 djeddit 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/EatEmAll/django-djedit/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_djedit
```


5.1 Development Lead

- Roman <megaclaff@gmail.com>

5.2 Contributors

- Logmytech <<https://github.com/logmytech/>>

Want to be on the list?

6.1 0.1.0 (2017-04-15)

- First release on PyPI.

6.2 0.1.5 (2017-05-26)

- Added tests for models, forms, views
- Urls that include topic titles with spaces in them don't work // fixed
- Competability fixes for Django 1.9 and 1.8 and Python 2.7
- client side fixes for added compatibility with mobile screen sizes
- font-awesome fonts are not loading // fixed
- Added description field for Topic model
- moved base.html to root template dir

6.3 0.2 (2017-07-04)

- Admins can lock/unlock & delete threads, delete posts
- Admins can edit thread title, url, and post content
- added users management page for admins
- admins can edit & delete topics
- added fields upvotes, downvotes, wsi (Wilson score interval) to Post model
- removed score field from Post model

- comments are ranked using wilson scoring interval
- added comments paging functionality (using “Load n more comments” links)

6.4 0.3 (2018-06-02)

- added option to define url prefix in settings.TOPICS_URL for topic & thread pages
- added sitemap support
- added url slugs to threads
- added ip & user agent tracking for posts
- added compatibility to Django 2.0
- added thread stickies (can be added or removed by admins)