
dj-termsearch Documentation

Release 0.2.0

James Cox

June 23, 2014

1	dj-termsearch	3
1.1	Documentation	3
1.2	Quickstart	3
1.3	Advanced	3
2	Installation	5
3	SingleTermSearchMixin	7
4	MultiTermSearchMixin	9
5	MapTermSearchMixin	11
6	Contributing	13
6.1	Types of Contributions	13
6.2	Get Started!	14
6.3	Pull Request Guidelines	14
6.4	Tips	15
7	Credits	17
7.1	Development Lead	17
7.2	Contributors	17
8	History	19
8.1	0.2.0 (2014-06-23)	19
8.2	0.1.0 (2014-06-20)	19

Contents:

dj-termsearch

Simple GET-based term searches for Django CBV's.

1.1 Documentation

The full documentation is at <https://dj-termsearch.readthedocs.org>.

1.2 Quickstart

Install dj-termsearch:

```
pip install dj-termsearch
```

Add "termsearch" to your INSTALLED_APPS then just add TermSearchMixin to a view and go:

```
from django.db import models
from termsearch.views import TermSearchMixin

class MyModelListView(TermSearchMixin, ListView):

    model = MyModel
    term = "title"
    lookup = "iexact"
```

Check the results at:

```
https://example.com/list?q=barry
```

1.3 Advanced

Use a list of model fields to use in the search:

```
class AnotherListView(TermSearchMixin, ListView):

    model = MyModel
    terms = ["title", "content", "author__name"]
    lookup = "iexact"
```

Map each field to a lookup type:

```
class YetAnotherListView(TermSearchMixin, ListView):

    model = MyModel
    term_mapping = {
        "title": "icontains",
        "tags__name": "iexact",
        "author__surname": "exact",
    }
```

Installation

At the command line:

```
$ easy_install dj-termsearch
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv dj-termsearch  
$ pip install dj-termsearch
```

SingleTermSearchMixin

Filter on a single model field.

Use the `string` representation of the field and the lookup type to filter the queryset:

```
from termsearch.views import SingleTermSearchMixin

class MyListView(SingleTermSearchMixin, ListView):

    model = MyModel
    term = "title"
    lookup = "iexact"
```

Raises an `ImproperlyConfigured` exception when missing the required attribute `term`.

`lookup` is optional and defaults to “exact” when not provided.

MultiTermSearchMixin

Filter on multiple model fields.

Use a `list` of model fields to use in the search:

```
from termsearch.views import MultiTermSearchMixin

class MyListView(MultiTermSearchMixin, ListView):

    model = MyModel
    terms = ["title", "author", "content"]
    lookup = "iexact"
```

Each term will filter the queryset using the `lookup` attribute.

Raises an `ImproperlyConfigured` exception when missing the required attribute `terms`.

As with `SingleTermSearchMixin`, `lookup` is optional and defaults to “exact” when not provided.

MapTermSearchMixin

Filter on multiple model fields with a lookup mapped to each.

Use a dict of model fields and lookups to use in the search:

```
from termsearch.views import MapTermSearchMixin

class MyListView(MapTermSearchMixin, ListView):

    model = MyModel
    term_mapping = {
        "title": "icontains",
        "tags__name": "iexact",
        "author__surname": "exact",
    }
```

Raises an ImproperlyConfigured exception when missing the required attribute `term_mapping`.

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/Ogreman/dj-termsearch/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

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

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

```
$ git clone git@github.com:your_name_here/dj-termsearch.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-termsearch
$ cd dj-termsearch/
$ 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 termsearch 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/Ogreman/dj-termsearch/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_termsearch
```

Credits

7.1 Development Lead

- James Cox <james.aaron.cox@gmail.com>

7.2 Contributors

None yet. Why not be the first?

History

8.1 0.2.0 (2014-06-23)

- Added separate search mixins (SingleTermSearchMixin, MultiTermSearchMixin, MapTermSearchMixin).
- Cleanup of docs.

8.2 0.1.0 (2014-06-20)

- First release on PyPI.