django-modeladmin-reorder Documentation

Release 0.2

Mishbah Razzaque

January 03, 2017

Contents

Contents:

CHAPTER 1

django-modeladmin-reorder

Custom ordering for the apps and models in the admin app. You can also rename, cross link or exclude models from the app list.

1.1 Features

- Reorder apps in admin index - this will allow you to position most used apps in top of the page, instead of listing apps alphabetically. e.g. `sites` app before the `auth` app
- Rename app labels easily for third party apps without having to modify the source code. e.g. rename `auth` app to `Authorisation` for the django admin app.
- Split large apps into smaller groups of models.
- Reorder models within an app. e.g. `auth.User` model before the `auth.Group` model.
- Exclude any of the models from the app list. e.g. Exclude `auth.Group` from the app list. Please note this only excludes the model from the app list and it doesn't protect it from access via url.
- Cross link models from multiple apps. e.g. Add `sites.Site` model to the `auth` app.
- Rename individual models in the app list. e.g. rename `auth.User` from `User` to `Staff`

1.2 Documentation

The full documentation is at https://django-modeladmin-reorder.readthedocs.org.

1.3 Install

Install django-modeladmin-reorder:

```
pip install django-modeladmin-reorder
```

1.4 Configuration

1. Add *admin_reorder* to *INSTALLED_APPS*:

```
INSTALLED_APPS = (
    ...
    'admin_reorder',
    ...
)
```

2. Add the *ModelAdminReorder* to *MIDDLEWARE_CLASSES*:

```
MIDDLEWARE_CLASSES = (
    ...
    'admin_reorder.middleware.ModelAdminReorder',
    ...
)
```

3. Add the setting *ADMIN_REORDER* to your settings.py:

```
ADMIN_REORDER = (
    # Keep original label and models
    'sites',

    # Rename app
    {'app': 'auth', 'label': 'Authorisation'},

    # Reorder app models
    {'app': 'auth', 'models': ('auth.User', 'auth.Group')},

    # Exclude models
    {'app': 'auth', 'models': ('auth.User', )},

    # Cross-linked models
    {'app': 'auth', 'models': ('auth.User', 'sites.Site')},

    # models with custom name
    {'app': 'auth', 'models': (
        'auth.Group',
        {'model': 'auth.User', 'label': 'Staff'},
    )},
)
```

CHAPTER 2

Installation

At the command line:

```
$ easy_install django-modeladmin-reorder
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv django-modeladmin-reorder
$ pip install django-modeladmin-reorder
```

CHAPTER 3

Usage

1. Add *admin_reorder* to *INSTALLED_APPS*:

```
INSTALLED_APPS = (
    ...
    'admin_reorder',
    ...
)
```

2. Add the *ModelAdminReorder* to *MIDDLEWARE_CLASSES*:

```
MIDDLEWARE_CLASSES = (
    ...
    'admin_reorder.middleware.ModelAdminReorder',
    ...
)
```

3. Add the setting *ADMIN_REORDER* to your settings.py:

```
ADMIN_REORDER = (
    # Keep original label and models
    'sites',

    # Rename app
    {'app': 'auth', 'label': 'Authorisation'},

    # Reorder app models
    {'app': 'auth', 'models': ('auth.User', 'auth.Group')},

    # Cross-linked models
    {'app': 'auth', 'models': ('auth.User', 'sites.Site')},

    # models with custom name
    {'app': 'auth', 'models': (
        'auth.Group',
        {'model': 'auth.User', 'label': 'Staff'},
    )},
)
```

CHAPTER 4

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

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

```
$ git clone git@github.com:your_name_here/django-modeladmin-reorder.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-modeladmin-reorder
$ cd django-modeladmin-reorder/
$ 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 admin_reorder 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/mishbahr/django-modeladmin-reorder/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_admin_reorder
```

CHAPTER 5

Credits

5.1 Development Lead

- Mishbah Razzaque <mishbahx@gmail.com>

5.2 Contributors

- zhangkaizhao - <zhangkaizhao@gmail.com>

CHAPTER 6

History

6.1 0.1.0 (2014-08-01)

- First release on PyPI.