

---

# **django-maintenance-window Documentation**

*Release 0.3.3*

**Jorik Kraaikamp**

**Sep 02, 2021**



---

# Contents

---

<b>1</b>	<b>django-maintenance-window</b>	<b>3</b>
1.1	Documentation . . . . .	3
1.2	Quickstart . . . . .	3
1.3	Settings . . . . .	4
1.4	Running Tests . . . . .	4
1.5	Credits . . . . .	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	0.2.1 (2017-10-30) . . . . .	15
6.2	0.2.0 (2017-10-26) . . . . .	15
6.3	0.1.0 (2017-10-26) . . . . .	15



Contents:



Your project description goes here

## 1.1 Documentation

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

## 1.2 Quickstart

Install django-maintenance-window:

```
pip install django-maintenance-window
```

Add it to your *INSTALLED\_APPS*:

```
INSTALLED_APPS = (  
    ...  
    'django-solo',  
    'django_maintenance_window',  
    ...  
)
```

Add django-maintenance-window's middleware to the middleware:

```
MIDDLEWARE_CLASSES = [  
    ...  
    'django_maintenance_window.middleware.MaintenanceModeMiddleware',  
    ...  
]
```

or

```
MIDDLEWARE = [  
    ...  
    'django_maintenance_window.middleware.MaintenanceModeMiddleware',  
    ...  
]
```

## 1.3 Settings

- **MAINTENANCE\_TEMPLATE = 'django\_maintenance\_window/maintenance.html'** Overwrite the template that is used for the maintenance template
- **MAINTENANCE\_DISPLAY\_END\_DATE = False** If the end date should be displayed at the bottom of the page.
- **MAINTENANCE\_EXCLUDE\_ADMIN\_URLS = True** This will allow accessing the admin even if maintenance mode is active.
- **MAINTENANCE\_EXCLUDE\_SUPER\_USER = False** This will allow super users to see the site even if maintenance mode is active.
- **MAINTENANCE\_EXCLUDE\_STAFF\_USER = False** This will allow staff users to see the site even if maintenance mode is active.

## 1.4 Running Tests

Does the code actually work?

```
source <YOURVIRTUALENV>/bin/activate  
(myenv) $ pip install tox  
(myenv) $ tox
```

## 1.5 Credits

Tools used in rendering this package:

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

## CHAPTER 2

---

### Installation

---

At the command line:

```
$ easy_install django-maintenance-window
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv django-maintenance-window  
$ pip install django-maintenance-window
```



To use `django-maintenance-window` in a project, add it to your `INSTALLED_APPS`:

```
INSTALLED_APPS = (  
    ...  
    'django_maintenance_window.apps.DjangoMaintenanceWindowConfig',  
    ...  
)
```

Add `django-maintenance-window`'s URL patterns:

```
from django_maintenance_window import urls as django_maintenance_window_urls  
  
urlpatterns = [  
    ...  
    url(r'^$', include(django_maintenance_window_urls)),  
    ...  
)
```



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

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

```
$ git clone git@github.com:your_name_here/django-maintenance-window.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-maintenance-window
$ cd django-maintenance-window/
$ 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 django_maintenance_window 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/jostcrow/django-maintenance-window/pull\\_requests](https://travis-ci.org/jostcrow/django-maintenance-window/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_django_maintenance_window
```



### 5.1 Development Lead

- Jorik Kraaikamp <jorik@maykinmedia.nl>

### 5.2 Contributors

None yet. Why not be the first?



### 6.1 0.2.1 (2017-10-30)

- Fixed that the admin would give a 500 when you only set the maintenance\_from.

### 6.2 0.2.0 (2017-10-26)

- Now the admin exclude will work with i18n urls.
- MAINTENANCE\_EXCLUDE\_ADMIN\_URLS is now a setting (default=True)
- MAINTENANCE\_EXCLUDE\_SUPER\_USER is now a setting (default=False)
- MAINTENANCE\_EXCLUDE\_STAFF\_USER is now a setting (default=False)

### 6.3 0.1.0 (2017-10-26)

- First release on PyPI.