
django*viewpermissions*

Release 0.0.2

Apr 10, 2020

Contents:

1	django-view-permissions	1
1.1	Overview	1
1.2	Installation	1
1.3	Usage	2
1.4	License	2
1.5	How To Contribute	2
1.6	Reporting Security Issues	2
1.7	Getting Help	3
2	How To Contribute	5
2.1	Installation	5
2.2	Updating the Requirements	5
2.3	Process	5
2.4	Support	6
3	Change Log	7
3.1	Unreleased	7
3.2	[0.0.2] - 19-03-2020	7
3.3	[0.0.1] - 19-03-2020	7
3.4	[0.0.dev0] - 17-03-2020	7
4	Indices and tables	9

django-view-permissions

django-view-permissions provides ways to control access for Django app views

1.1 Overview

Checks the incoming requests, to grant or deny access for views according to the permissions defined in the views. If permissions attribute is not defined in the view, normal django flow is followed. If permissions attribute is defined in view it checks and grants or denies access of the view.

1.2 Installation

Install the latest release using pypi:

```
pip3 install django-view-permissions
```

Add the application to the INSTALLED_APPS:

```
INSTALLED_APPS = (  
    ...  
    # DjangoViewPermissions  
    'django_view_permissions',  
)
```

Add the middleware to the configuration:

```
MIDDLEWARE_CLASSES = (  
    ...  
    'django_view_permissions.middleware.DjangoViewPermissionsMiddleware',  
)
```

1.3 Usage

Currently, 3 ways to define permissions are supported.

- Attribute based permissions
- Method based permissions
- Class based permissions

Below is a Attribute based permission example. Where view will be only accessible to users whose attribute 'is_staff' is 'True'.

```
class DummyView(View):
    permissions = [
        ('attr', 'is_staff', True),
    ]
```

In Method based permissions, you will need to define the method with a 'request=None' argument and a boolean return statement. All the requests for which method returns 'True' will be able to access the view.

```
class DummyView(View):
    permissions = [
        ('method', permission_method),
    ]
```

In Class based permissions, you will need to define a class with a 'request=None' argument in __init__ method. All the requests for which __call__ method returns 'True' will be able to access the view.

```
class DummyView(View):
    permissions = [
        ('class', PermissionClass),
    ]
```

Note: It is recommended to define the permission methods or classes in a separate file.

Please check [permissions.py](#) for examples.

1.4 License

The code in this repository is licensed under the Apache Software License 2.0 unless otherwise noted.

Please see `LICENSE` for details.

1.5 How To Contribute

Contributions are very welcome.

Please read [How To Contribute](#) for details.

1.6 Reporting Security Issues

Please do not report security issues in public. Please email muhammadayubkhan6@gmail.com.

1.7 Getting Help

Feel free to create git issues in case of queries/issues/enhancements.

2.1 Installation

Fork the repository and run:

```
$ git clone git@github.com:<username>/django-view-permissions.git
$ cd django-view-permissions
$ make venv # needs python3.5
$ make requirements
```

2.2 Updating the Requirements

pip-tools is used for the requirement management. If you want to add/update some existing requirement, you need to update it in the .in file and run the following command.

```
$ make upgrade
```

2.3 Process

1. Share the idea. (Create a Issue)
2. Submit first working draft of your idea. (Create a Pull Request)
3. Incorporate the reviewers suggestions.
4. Get a couple of thumbs up and merge. (Y)

2.4 Support

For any kind of queries feel free to create github issues.

3.1 Unreleased

3.1.1 Added

- _

3.2 [0.0.2] - 19-03-2020

3.2.1 Added

- Fix some documentation and release issues.

3.3 [0.0.1] - 19-03-2020

3.3.1 Added

- Automatic upload to PyPI on tags.
- Documentation

3.4 [0.0.dev0] - 17-03-2020

3.4.1 Added

- Attr based permissions

- Method based permissions
- Class based permissions
- First release on PyPI.

CHAPTER 4

Indices and tables

- `genindex`
- `modindex`
- `search`