

---

# Django MFA Documentation

*Release 1.0*

**Micro Pyramid**

**May 01, 2018**



---

# Contents

---

<b>1</b>	<b>Getting started</b>	<b>3</b>
1.1	Requirements . . . . .	3
1.2	Installation . . . . .	3
1.3	Settings . . . . .	3
1.4	Urls . . . . .	3
<b>2</b>	<b>Flow And Urls</b>	<b>5</b>
2.1	Step1: Enabling MFA . . . . .	5
2.2	Step2: Setup with APP . . . . .	5
2.3	Step3: Scan QR code . . . . .	5
2.4	Step4: Disabling the MFA . . . . .	6
<b>3</b>	<b>Contributing</b>	<b>7</b>
3.1	Sending pull requests . . . . .	7



This application provides an implementation of Multi-Factor Authentication.

The **source code** for this app is hosted at <https://github.com/MicroPyramid/django-mfa>

Contents:



### 1.1 Requirements

Python	>= 2.6 (or Python 3)
Django	< 1.10

### 1.2 Installation

The Git repository can be cloned with this command:

```
git clone https://github.com/MicroPyramid/django-mfa
```

The `django_mfa` package, included in the distribution, should be placed on the `PYTHONPATH`.

Otherwise you can just `easy_install -Z django-mfa` or `pip install django-mfa`.

### 1.3 Settings

Add `django_mfa` to the `INSTALLED_APPS` to your `settings.py`.

See the settings section for other settings.

Add `'django_mfa.middleware.MfaMiddleware'` to the `MIDDLEWARE_CLASSES`

### 1.4 Urls

Add the following to your root `urls.py` file.

```
urlpatterns = [  
    ...  
    url(r'^settings/', include('django_mfa.urls', namespace="mfa")),  
]
```

Done. With these settings you have now, you will get the MFA features.



Once we have done with the basic setup as given in installation and setup, we are ready to go.

### 2.1 Step1: Enabling MFA

Once your app is up and running with settings, just go to url `/settings/security`. There we can see if the MFA is enabled or not. Lets say MFA is not enabled. When we click on 'Enable MFA'. Then user will be redirected to setup types url.

### 2.2 Step2: Setup with APP

When user completes step1 successfully, he/she will be redirected to url `/settings/mfa/configure/`. Here user will have option to Setup using APP. This means we are going to use google authenticator for this. When user click on 'Setup using APP' then will be redirected to page where we get a QR code which we have to scan in our google authenticator.

### 2.3 Step3: Scan QR code

In this step user will see a QR code generated with user's email. User have to scan this code with Google Authenticator app. Once user scanned this image, user will get a 6 digit code which expire for every 30 secs. User has to enter this six digit code in the text box. Once user successfully enters the six digit code, thats it from now onwards your MFA is enabled. You can check this by logging out of the system and login again. Even when user login with username and password the application will ask for 6 digit code to complete the login.

## 2.4 Step4: Disabling the MFA

Disabling MFA is quite simple, we can do this by visiting the url `/settings/security`. By clicking the 'Disable MFA' your account will be free from MFA.

Feel free to create a new Pull request if you want to propose a new feature or fix a bug.

### 3.1 Sending pull requests

1. Fork the repo:

```
https://github.com/MicroPyramid/django-mfa
```

2. Create a branch for your specific changes:

```
$ git checkout master  
$ git pull  
$ git checkout -b feature
```

To simplify things, please, make one branch per issue (pull request). It's also important to make sure your branch is up-to-date with upstream master, so that maintainers can merge changes easily.

3. Commit changes. Please update docs, if relevant.
4. Don't forget to run tests to check than nothing breaks.
5. Ideally, write your own tests for new feature/bug fix.
6. Submit a pull request.