

Django Files And How They Work

🕒 Created	@October 6, 2023 9:38 PM
☰ Made By	Mariam Tamer

What I'll discuss here :

- 1) Recap on last file, how to create any Django project.
- 2) Your created project built in files.
- 3) How Django works.

1) Let's recap our steps to create any Django project and run it on server:

- 1) Install Pythonv3.8 or newer
- 2) Install Virtual Environment
- 3) Create Virtual environment
- 4) Activate Virtual Environment
- 5) Install Django
- 6) Create Project
- 7) Run the project on server & migrate

On visual Studio Code, run your server :

python3 manage.py runserver

2) Now let's get to know the new installed built in files :

1) __init__.py

2) asgi.py

3) settings.py

4) urls.py

5) wsgi.py

6) db.sqlite3

1) __init__.py :

Contains your imported libraries, and you won't use it much.

2) asgi.py && 5) wsgi.py :

-Files related to server.

-You won't use them.

3) settings.py

The brain of the project. Contains everything.

- Contains your **SECRET_KEY** that you shouldn't share with anyone.
- Contains your **installed apps** and any app you install ,you should write it here :

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
]
```

- Contains your **Frontend templates** :

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [],  
        'APP_DIRS': True,  
        'OPTIONS': {  
            'context_processors': [  
                'django.template.context_processors.debug',  
                'django.template.context_processors.request',  
                'django.contrib.auth.context_processors.auth',  
                'django.contrib.messages.context_processors.messages',  
            ],  
        },  
    },  
]
```

-Contains your **used databases**. It has **sqlite3** so it could work, but you can use **PostgreSQL** for big projects.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

-Contains **password validation** :

```
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]
```

-Contains **static url** or files such as **html/css/images**

`STATIC_URL = 'static/'`

4) urls.py

-Very important too, and you'll be using it for all your created paths.

-If you wanna create a path like :

facebook.com/groups

facebook.com/settings

facebook.com/profile

and so on.. you'll write them and create them there ...

```
from django.contrib import admin
from django.urls import path

urlpatterns = [
    path('admin/', admin.site.urls),
]
```

***** ((3)) How Django Works ? *****

-By creating **web applications** from your project.

-**Project** makes **many apps**. Each app is responsible for a path.

(Like I gave you the Facebook example before : groups - settings - profile) ,

your django project **creates apps, one for each path** .. one for groups , another for settings, another for profile

-Each app contains the following :

- **Views**
- **Models**
- **Templates**
- **Urls**

Views :

-The leader of them all.

-Decides what is displayed, and what is not.

Models :

-Your **Database**

Templates :

-You create this.

-This is your created **Frontend**.

Urls :

-You create this too.

-This is different than the one in the project built in files.