

Blog Project

Zofia Dobrowolska

9 marca 2023

1 Create virtual environment

Project created for deployment needs a dedicated virtual environment to run in production. Step-by-step instruction on how to do it for Django project.

1. Navigate to root of a project:

```
~ cd <<project path>>
```

2. Create the virtual environment .venv:

```
~ python -m venv <<name of the virtual environment>>
```

3. Activate the venv:

```
~ venv\Script\activate
```

4. Install django:

```
~ pip install django
```

5. Create the Django project:

```
~ django-admin startproject <<name>>
```

6. Check by running the server (this command starts the project, making it accessible from web):

```
~ python manage.py runserver
```

7. Add an app to the project:

```
~ django-admin startapp <<name>>
```

8. Migrate changes:

```
~ python manage.py makemigrations
~ python manage.py migrate
```

9. Create an admin (user):

10. Migrate changes:

```
~ python manage.py createsuperuser
```

Then you will be asked to enter the user name, password, and email address. You can omit the email, though.

11.

2 Handling venv

Storing a Django project and its virtual environment objectively consumes too much space. Yet, for deployment, you have to know the requirements for the project to work. You can generate *requirements.txt* by typing in git BASH a command¹:

```
env1/bin/python -m pip freeze \> requirements.txt
```

and to install from it to another environment run command:

```
env2/bin/python -m pip install -r requirements.txt
```

Then include *.venv* in your *.gitignore* file.

¹[pip freeze](#)

3 .gitignore

The *.gitignore* file is a text file that tells Git which files or folders to ignore in a project². A local *.gitignore* file is usually placed in the root directory of a project, but you can also create a global one. Mind that, entries in the global file will be ignored in all your Git repositories.

To create a local *.gitignore* file, create a text file and name it *.gitignore* (remember to include the *.* at the beginning). Then edit this file as needed. Each new line should list an additional file or folder that you want Git to ignore.

The entries in this file can also follow a matching pattern.

- *** is used as a wildcard match
- */* is used to ignore pathnames relative to the *.gitignore* file
- *#* is used to add comments to a *.gitignore* file

Par exemple:

```
# Ignore Mac system files
.DS_store

# Ignore node_modules folder
node_modules

# Ignore all text files
*.txt

# Ignore files related to API keys
.env

# Ignore SASS config files
.sass-cache
```

To add or change your global *.gitignore* file, run the following command:

To add or change your global *.gitignore* file, run the following command:

This will create the file *~/.gitignore_global*. Now you can edit that file the same way as a local *.gitignore* file. All of your Git repositories will ignore the files and folders listed in the global *.gitignore* file.

To untrack a single file, ie stop tracking the file but not delete it from the system use:

²[.gitignore](#)

```
git rm --cached filename
```

To untrack every file in .gitignore:

First commit any outstanding code changes, and then run:

```
git rm -r --cached
```

This removes any changed files from the index(staging area), then run:

```
git add .
```

Commit it:

```
git commit -m ".gitignore is now working"
```

To undo

```
git rm --cached filename
```

, use git add filename