**Django project with Docker and PostgreSQL**

By default, a django built-in project comes with an integrated SQLite3 database and a default configuration which in most cases is unappropiated for what we want to do. In this document, I’ll set out how to configure your Django project by using the templates and settings provided by Django. Also, I’ll explain how to configure your database switching from SQLite3 to PostgreSQL and how to integrate your application in a Docker container to keep all the dependencies in the same container and make easier its future deploy.

* First of all, we create the directory for our code, and then install django. Also, as we are using PostgreSQL, psycopg2 installation is needed.

1. mkdir directory && cd directory
2. pipenv install django~=3.1.0 psycopg2-binary==2.8.5
3. pipenv shell

I’ll name the django project config

1. django-admin startproject config . (the period is important to avoid the creation of an extra directory by Django)
2. python manage.py runserver
3. go to: 127.0.0.1:8000
4. type *exit* to turn off the virtual environment

* At second, Docker’s set up is needed:

1. To install Docker read the docs: <https://docs.docker.com/get-docker/>
2. Create two files inside your root directory: *Dockerfile* and *docker-compose.yml*
3. Dockerfile content:
4. FROM python:3.9
5. ENV PYTHONDONTWRITEBYTECODE 1
6. ENV PYTHONUNBUFFERED 1
7. WORKDIR /directory
8. COPY Pipfile Pipfile.lock / directory /
9. RUN pip install pipenv && pipenv install --system
10. COPY . / directory /

directory represents tour root directory

1. Docker-compose content
2. version: '3.3'
3. services:
4. web:
5. build: .
6. command: python3 / directory /manage.py runserver 0.0.0.0:8000
7. volumes:
8. - .:/code
9. ports:
10. - 8000:8000
11. depends\_on:
12. - db
14. db:
15. image: postgres:11
16. volumes:
17. - postgres\_data:/var/lib/postgresql/data/
18. environment:
19. - "POSTGRES\_HOST\_AUTH\_METHOD=trust"
20. volumes:
21. postgres\_data: