Crear un proyecto de django

1. Crear un entorno virtual

Python -m venv {nombre del entorno}

1. Activar entorno virtual

{nombre del entorno}/script/activate

1. Instala django

Pip install django

1. Instalar unipath

Pip install unipath

1. Installar psycopg2 (es el que maneja el postgres)

pip install psycopg2

1. Crear el proyecto

django-admin startproject {nombre del proyecto}

Postgresql

1. Después de instalar se debe ingresar al Shell

Poner por defecto el server, database, port useranme y en contraseña se pone la contraseña que configuraron al principio.

1. Crear database

CREATE DATABASE {nombre bd};

1. Crear usuario

CREATE USER {usuario};

1. Dar permisos del usuario a la base de datos.
2. Ingresar a la base de datos

\c {base de datos}

1. Alterar la bd para dar el permiso al usuario

ALTER ROLE {usuario} WITH PASSWORD ‘{contraseña}’;

1. Alterar la bd por un error que sale si no se hace este paso
2. Crear INDEX en la base de datos para hacer consultas con trim desde django
   1. Entrar a la Shell de potsgresql
   2. Ingresar a la base de datos usada con el comando \c {nombre bd}
   3. Crear extensión con CREATE EXTENSION pg\_trgm;
   4. Crear index CREATE INDEX {nombre como se va a llamar el index}\_idx ON {nombre de app\_nombre del modelo} USING GIN({campo de la tabla} gin\_trgm\_ops);

EJ: CREATE INDEX libro\_titulo\_idx ON libro\_libro USING GIN(titulo gin\_trgm\_ops);

ALTER DATABASE {nombre bd} OWNER TO {usuario};

NOTAS

El los archivos manage.py y wsgi.py se debe de cambiar la ruta nueva del archivo de settings.local

Cuando se creen carpetas con código se debe crear un archivo llamado \_\_init\_\_.py

COMANDOS IMPORTANTES CMD

1. Migrar base de datos

Python manage.py makemigrations

Python manage.py migrate

1. Crear usuario en el proyecto

Python manage.py createsuperuser

1. Correr el servidor

Python manage.py runserver

1. Crear apps

django-admin startapp {nombre app}