**DJANGO**

**Usamos: Rest Framework y Postman (para simular peticiones, simula el front)**

Pip install django

Pip install djangorestframework

pip install --only-binary :all: mysqlclient

django-admin startproject heroesApp para crear el proyecto

Python manage.py runserver para levantar el proyecto

Ctrl + C para parar el servidor

Manage.py startapp <nombre de la app> para crear una app

Las apps dividen el proyecto

Python manage.py migrate aplica la migracion

Python manage.py makemigrations crea la migración

Python manage.py createsuperuser para crear un usuario

Admin

Milimolinaa@hotmail.com

mjmjulieta

* Para utilizar otro motor de base de datos distinto a sqlite

En settings -> datebases -> engine

También debo modificar el puerto, usuario, nombre de la base y contraseña

Después instalo el conector del motor elegido (pip install mysqlclient)

* Modelos

Se definen con clases

Dentro del archivo models.py

Debo definir la aplicación en settings

Luego del migrate se creará la tabla correspondiente en la base

Para ver la tabla en Admin, debo agregar en el archivo admin.py

admin.site.register(<modelo>)

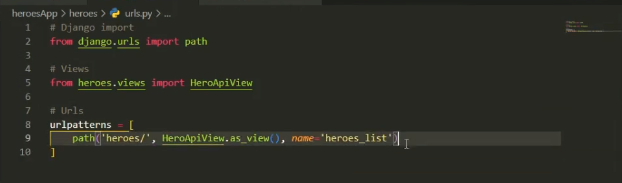
Los servicios se crean en views.py y heredan de APIView (importado de rest\_framework.views)



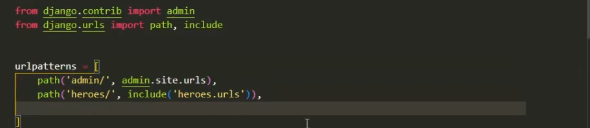
* Peticiones con postman

Necesito una URL

Creo un archivo urls.py dentro de la carpeta de la aplicación



Y luego agrego los include en el urls.py que ya existía



Las respuestas deben ser del tipo https response

Para esto importo “from rest\_framework.response import Response”

Creo la view con funciones de las peticiones

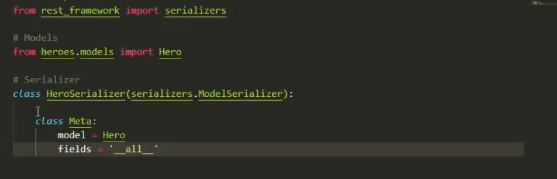
Para que el front y el back se comuniquen necesito los serializers



* Para conectar la base de datos a las peticiones

Tengo que serializarlas para pasar de queryset a diccionario

Utilizo un archivo llamado serializer.py



**REACT**

Npm start iniciar el server

Npx create-react-app <nombre> crear la app

npm i --save react-select

npm i react-router-dom