Django Dev

Ejercicio 1

Armar una API REST en Django con integración con Django Rest Framework que guarde las **mediciones** de consumo de energía de los distintos **medidores** en una base de datos relacional.

De los medidores nos interesa conocer su llave identificadora (código alfanumérico único) y nombre. Por otro lado, de las mediciones nos interesa conocer la fecha hora, el consumo registrado (*kwh*) y el medidor que registro dicha medición. Los valores de consumo de las mediciones deben ser siempre positivos, las mediciones con valores negativos deben ser descartadas.

Se debe desarrollar un endpoint que permita dar de alta nuevos medidores enviando su llave y nombre.

Mediante un endpoint los medidores enviaran las mediciones (una por una) especificando los datos previamente mencionados. **No se es necesario implementar ningún tipo de script que simule a los medidores, solamente se debe desarrollar el endpoint que reciba las mediciones.**

Por último, se deben crear los siguientes endpoints que permitan obtener datos de las mediciones previamente ingresadas:

- Máximo Consumo: Este endpoint deberá retornar la medición con consumo máximo de un medidor en específico.
- Mínimo Consumo: Este endpoint deberá retornar la medición con consumo mínimo de un medidor en específico.
- Consumo total: Este endpoint deberá retornar el consumo total de un medidor, el consumo total puede entenderse como la sumatoria de consumo de todas sus mediciones. Ej: Si dispongo de tres mediciones asociadas al Medidor 1: [12kw, 20kw, 87kw] el consumo total sería: 119kw.
- Consumo promedio: Este endpoint deberá devolver el consumo promedio de un medidor en específico.

Se deben documentar los endpoints con la información necesaria para poder probar y testear su funcionamiento. La solución debe ser de fácil de replicar y desplegar.

Consideraciones

- Debe estar subido a Github
- Opcionalmente se puede dockerizar la solución.
- Usar sqlite
- No es necesario utilizar autenticación

Además del código mándanos un texto explicando alguna decisión de diseño particular que hayas tomado y qué simplificaciones hiciste por ser un ejemplo de prueba.

Ejercicio 2

Partiendo del siguiente código y utilizando la **menor cantidad de líneas**, resuelva los siguientes puntos:

```
import json

repetidos = [1,2,3,"1","2","3",3,4,5]

r = [1,"5",2,"3"]

d_str = '{"valor":125.3,"codigo":123}'
```

1. Genere una lista con los valores no repetidos de la lista '*repetidos'*.
2. Genere una lista con los valores en común entre la lista '*r*' y '*repetidos'*
3. Transforme '*d_str*' en un diccionario.

Este ejercicio puede resolverse en una script independiente del ejercicio 1.