

Correcta Organización de Carpetas

Dentro de la carpeta requirements voy a alojar todos los requerimientos que va a necesitar mi proyecto para ejecutarse

- Para ver los paquetes instalados en mi entorno local uso pip freeze --local
- Para instalar los paquetes desde el txt uso: pip install -r .\local.txt

Archivo Secret, para esconder información sensible en el proyecto

A fin de poder esconder información sensible tales como los entornos donde esta la conexión a la BD, pues se usa un archivo JSON secret a la altura del manager

En este archivo estará la info sensible a la cual yo llamaré desde cualquier archivo siempre que sea necesario y al ser un diccionario pues le llamaría por su clave respectivamente

Secret.json y Gitignore

Se crea un Gitignore donde especifico los archivos que no quiero que se suban al repositorio por seguridad

Aplicaciones y AbstractBaseUser

Siempre que se trabaje con usuarios el nombre de aplicación se recomienda que sea users

Para el caso de usuarios Django nos facilita una forma sencilla para crear los modelos haciendo uso de la clase interna AbstractUser

Sumado a esto para poder tener un control más robusto de todos los usuarios de Django lo que se utiliza es una clase llamada PermissionsMixin

La ventaja de usar esta clase es que le indicamos a Django que únicamente trabaje con el modelo creado por nosotros y no con el interno, además de poder hacer un seguimiento de todos los superusuarios

Se debe configurar en el base.py indicando que ahora voy a trabajar con otro modelo

Además al redefinir el modelo Usuario debo indicarle a Django cuáles son las variables o atributos de autenticación.

Para ello lo especifico en el modelo que atributo hace referencia a USERNAME_FIELD

Sin embargo también se requiere de managers para que funcione correctamente lo que se hará en la siguiente clase

Escribiendo User Managers

En mi archivo managers.py importo el BaseUserManager

A fin de poder indicar si el usuario que se crea es superusuario y si puede acceder al admin lo que se haria es redefinir el modelo sin embargo eso ya lo trae el AbstractBaseUser

