

¿CÓMO FUNCIONA LA AUTENTICACIÓN Y FLASK LOGIN?

Amigos espero que este pequeño resumen les sea de ayuda.

Básicamente lo que estamos haciendo es:

- A través de **UserData** almaceno los datos de usuario y contraseña (cada vez que haya una instancia de esta clase va a tener esos dos atributos).
- Creo la clase **UserModel** que extiende la clase **UserMixin**: Esto lo hago para tener acceso a los métodos que ya trae por defecto esta clase (**como el método para autorizar**). Además en el constructor le creo un **id** y un **password** (que ya tengo guardados en la clase **UserData**).
- Creo el **método query** que sirve para pedir información en la base de datos con el **método que creé en firestore_service.py** `get_user`.

AMPLIACIÓN: El **método get_user** recibe como parámetro el **id** que necesite cuando haga el llamado de este método y me devuelve de la colección **users** el documento correspondiente al **user_id** que le pasé. Después de hecho el proceso de obtener este documento, ahora creo una instancia de **UserData** con el documento que obtuve. Lo cual me sirve para ahora si hacer la autenticación, haciendo uso de la función **load_user** que creé en el **__init.py__**.

```
@login_manager.user_loader
def load_user(username):
    return UserModel.query(username)
```

El decorador `@login_manager.user_loader` lo que hace es usar la librería flask-login que ya había importado previamente, y con el método **user_loader** de esa librería recargo el **id** del usuario que ya tengo almacenado en la sesión para devolver el objeto correspondiente a ese usuario.

FIN: Con el objeto que obtuvimos ya estamos listos para autenticarlo.

NOTA: Todo esto se hace para darle al usuario una forma fácil de hacer login y logout, proteger sus datos, darle una vista restringida a los usuarios que hagan login o logout y algunas funciones mas que trae **flask login**.

Espero no haberlos confundido mas y que por el contrario les haya sido de ayuda.

Saludos.