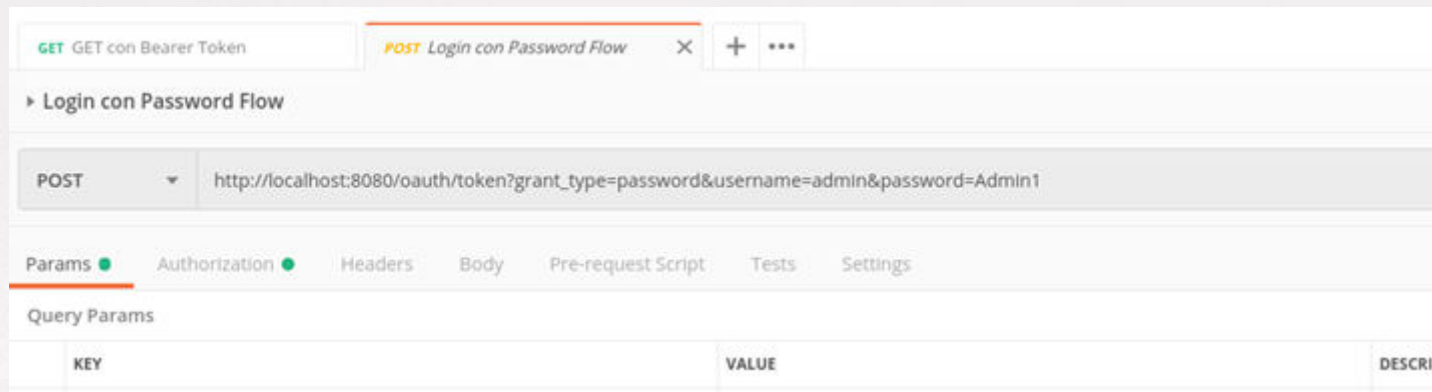


# OAuth2: Ejecución de nuestra aplicación

Implementa la seguridad de tu API Rest con Spring Boot

# Pruebas con Postman

- Probamos el flujo de tipo Password
- *client\_id* y *client\_secret* como autenticación básica



# Pruebas con Postman

- Podemos usar el Bearer token para hacer otras peticiones

```
1 {  
2   "access_token": "517d0695-3ed4-4407-b3b0-f96a9e4f513d",  
3   "token_type": "bearer",  
4   "refresh_token": "5119940f-f0ec-4b73-991e-df522217fa06",  
5   "expires_in": 84862,  
6   "scope": "read"  
7 }
```

# Pruebas con Postman

The screenshot shows the Postman interface for a GET request to `http://localhost:8080/producto/5`. The request is configured with a Bearer Token authorization header. The response is a JSON object with the following structure:

```
1 {
2   "id": 5,
3   "nombre": "Wine - Beringer Founders Estate",
4   "precio": 27.0,
5   "imagen": "http://dummyimage.com/139x103.bmp/5fa2dd/ffffff",
6   "categoria": {
7     "id": 2,
8     "nombre": "Bebida"
9   },
10  "lotes": []
11 }
```

The status of the request is 200 OK.

# Pruebas con Postman

- Hemos añadido el controlador `/user/me` de otros proyectos de ejemplo, que nos da la información del perfil de usuario en función de su token.

```
Pretty Raw Preview Visualize BETA JSON 
1 {
2   "username": "admin",
3   "avatar": "https://api.adorable.io/avatars/285/admin@openwebinars.net.png",
4   "fullName": "Admin admin",
5   "email": "admin@openwebinars.net",
6   "roles": [
7     "ADMIN",
8     "USER"
9   ]
10 }
```

# Aplicación cliente

- Probamos la implementación de un cliente Javascript muy sencillo que nos permita probar el flujo *implícito* y el *password*.
- Ejecutamos el proyecto y cargamos en el navegador `http://localhost:8081`

