Especialización en Back End II

Flujo de autenticación de OpenID Connect

A continuación, veremos cómo podemos recrear un flujo de autenticación manualmente para lograr autenticar a un usuario.

Vamos a realizar una consulta mediante Postman al siguiente endpoint (tené en cuenta que las zonas marcadas con negrita pueden variar los datos dependiendo de qué tipo de configuración hayas hecho en tu PC) y vamos a analizar la respuesta que nos proporciona:

- **http://localhost**:**8080**/realms/**prueba-reino**/.well-known/openid-configuration

Podemos ver dentro de la respuesta el campo **authorization_endpoint**.


```
{
  "issuer": "http://localhost:8080/realms/prueba-reino",
  "authorization_endpoint": "http://localhost:8080/realms/prueba-reino/protocol/openid-connect/auth",
  "token_endpoint": "http://localhost:8080/realms/prueba-reino/protocol/openid-connect/token",
  "introspection_endpoint": "http://localhost:8080/realms/prueba-reino/protocol/openid-connect/token/introspect",
  "userinfo_endpoint": "http://localhost:8080/realms/prueba-reino/protocol/openid-connect/userinfo",
  "end_session_endpoint": "http://localhost:8080/realms/prueba-reino/protocol/openid-connect/logout",
```


Ahora utilizaremos esa URL que nos devuelve para crear una **request de autenticación**, que luego nos redirigirá a **Keycloak** para ingresar las credenciales del usuario.

La URL tiene la siguiente estructura básica:

http://localhost:8080/realms/**prueba-reino**/protocol/openid-connect/auth

Pero, además de ello, existen una serie de parámetros que podemos especificar:

- **client_id**: el ID del cliente que sea creado en Keycloak.
- **response_type**: indica que queremos que retorne un código (este código luego lo usamos para cambiarlo por un token).
- **redirect_uri**: en un caso real, esta URL debería ser la de nuestra aplicación front-end o el lugar al que queramos redirigir la app luego de la autenticación, en nuestro caso,

es solo a modo de ejemplo.

- **scope**: los scopes se utilizan para especificar qué tipo de acceso solicitamos cuando realizamos una petición.

Un **ejemplo** de cómo aplicarlo podría ser este:

- **http://localhost:808**0/realms/**prueba-reino**/protocol/openid-connect/auth?**client_id**=oidc-postman&**response_type**=code&**redirect_uri**=http://localhost:8080/&**scope**=openid

Si abrimos esta URL en el navegador, podemos ver que somos redirigidos a la pantalla de inicio de sesión de Keycloak. Donde se nos pide un usuario y contraseña para poder ingresar.

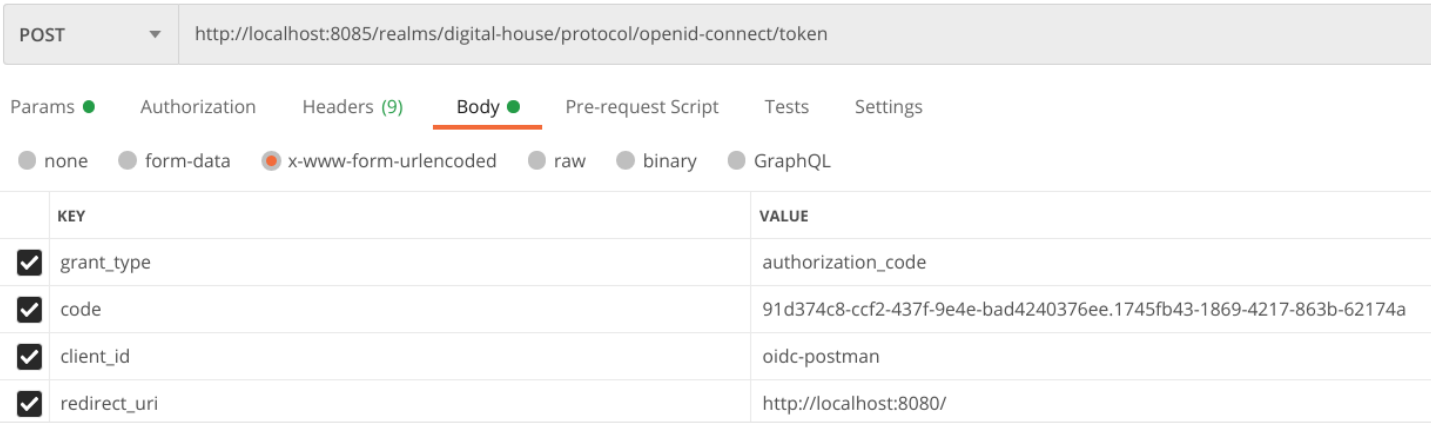
Luego de ingresar un usuario y contraseña válidos, Keycloak nos retorna el código que le solicitamos. Este código lo podemos tomar de la URL a la que nos redirige. En nuestro caso es:

- http://localhost:8080/?session_state=7e04cd1c-780c-41d4-bb44-a6b1d0746876&code=**1f381d5e-46da-4942-b204-cc64fdbca4fe.7e04cd1c-780c-41d4-bb44-a6b1d0746876.dcddbad9-696a-4b1a-a070-6af5c14a586f**

En donde el valor de **code** es el código que nos interesa tomar, para intercambiar por un token.

Obteniendo el token de un usuario autenticado

Si volvemos a la respuesta de la consulta al endpoint **http://localhost:8085**/realms/**digital-house**/.well-known/openid-configuration, podemos ver el campo **token_endpoint**, que tiene la URL que necesitamos para conseguir el token del usuario autenticado. Desde Postman enviamos el siguiente request:



Y obtenemos:


```
{
    "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJPd2lJbENBMU1
    "expires_in": 300,
    "refresh_expires_in": 1800,
    "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICIxZTFlY2I4OS
    "token_type": "Bearer",
    "id_token": "eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJPd2lJbENBMU1XZEJ
    "not-before-policy": 1651034163,
    "session_state": "1745fb43-1869-4217-863b-62174a2edbe9",
    "scope": "openid email profile"
}
```


Si obtenemos algún error de tipo **invalid_grant**, puede ser debido a las siguientes razones:

- Realizamos la petición para obtener el token demasiado lento. El código de autorización solo es válido durante 1 minuto. Tendremos que reintentarlo más rápido.
- Enviamos la misma petición más de una vez. El código de autorización es válido para usar una única vez.