Keycloak book

jueves, 12 de agosto de 2021 19:32

oAuth 1º

Resource owner	Resource server	Client	Authorization Server (Keycloak)

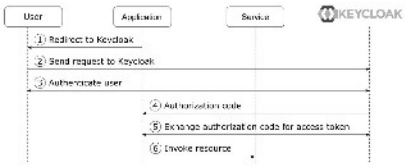
El cliente solicita al servidor de autorización el acceso a un recurso en nombre del propietario del mismo.

Flujos (Grant types)

- Client credentials: Si la app está accediendo al recurso en su propia nombre
- Device flow: si la app no tiene navegador ni forma de insertar texto (smartTV)
- Authorization code flow: si ninguno de arriba aplica, usa este.

2 flujos "legacy"

- Implicit Flow
- Resource owner Password Credentials flow



Dentro del flujo de oAuth hay 2 clients:

- Confidential: server-side applications capaces de almacenar un secreto.
- Public: Client-side apps que no son capaces de almacenar información de forma segura. No pueden autenticarse con el servidor de autenticación, hay 2 opciones:

① El servidor de autenticación sólo enviará un código de autenticación a la aplicación hospedada en una URL pre-configurada.

② Proof Key for Code Exchange (PKCE)
Es una extensión de oAuth 2.0 previene que alguien intercepte un código de auth de ser intercambiado por un access token.

Especificaciones adicionales

- Bearer Token: oAuth 7.0 no describe el tipo de accessToken o cómo debe usarse. Habitualmente expresa en la cabecera de Authorization.
- Token Introspection: El contenido

OIDC 2º

oAuth 7.0 no cubre la autenticación

Roles en OIDC:

End User	Relying Party (app in oAuth) RP	OpenID Providers (Keycloak) OP

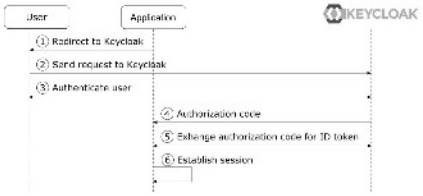
- OIDC se construye por encima de oAuth
- Utiliza Authorization Code grant type de oAuth 2.0. La diferencia es que el cliente incluye scope=openid en la petición inicial.

Flujos

- Authorization code flow: Usa el mismo flujo que oAth code grant type y retorna un código de autorización para ser intercambiado por: IDToken, AccessToken, RefreshToken
- Hybrid flow: El idToken se devuelve a la vez que el código de autenticación en la petición inicial.

Flujos legacy

- Implicit Flow



Especificaciones adicionales

- Discovery: Permite a los clientes descubrir información sobre el OP.
- Dynamic Registration: Permite a los clients registrarse ante el OP.
- Session Management: Define como monitorizar la sesión de usuario con el OP y cómo el cliente puede iniciar un cierre de sesión.
- Front-Channel Logout: Usando iframes embebidos define un mecanismo para SSOut de multiples aplicaciones.
- Back-Channel Logout: Define un mecanismo para SSOut para múltiples apps usando un mecanismo de petición través entre el RP (app).

OIC tiene 2 conceptos adicionales por encima de oAuth 7.0.

KEYCLOAK 3º

Keycloak delega en JWT el formato de su IDToken y del AccessToken.

OIDC tiene 2 conceptos adicionales por encima de oAuth 2.0.

- Especifica el formato del IDToken en JWT (claro y no opaco). Tiene un formato bien especificado y contiene los valores (claims) directamente en el token.

- Define un **UserInfo Endpoint** que puede ser invocado con un accessToken y devuelve los mismos claims que el IDToken.

TIP: Pueden haber actualizado desde la emisión del AccessToken y traer info nueva.

---

usarse. Habitualmente [illegible] cabecera de Authorization.

- **Token Introspection**: El contenido del AccessToken es opaco a las apps, el token introspection endpoint permite a los clientes obtener información del access Token.

- **Token Revocation**: oAuth 2.0 considera cómo accessToken es emitido a las aplicaciones, pero no cómo son revocados. Para eso es este endpoint.