OpenID Connect (OAuth 2.0 + Autenticação federada UA)

O consumo de recursos, por exemplo métodos de uma API, baseia-se na obtenção de autorização para consumo desses recursos e na identificação do utilizador que está a solicitar essa autorização.

A obtenção de autorização assenta na obtenção de um token de acesso (access token), podendo, no entanto, ser obtida informação adicional (ID token) relativamente ao utilizador (autenticado) a quem a autorização está a ser atribuída.

Há vários métodos de obtenção de um access token, sendo que eles podem ser divididos em termos de interação:

- Necessidade/possibilidade de um utilizador interagir com um user agent (browser):
  - Authorization Code (OAuth 2.0)
  - Implicit (OAuth 2.0)
- A comunicação é feita entre sistemas.
  - Client Credentials (OAuth 2.0)
  - Password (OAuth 2.0)

Relativamente ao ID token, no contexto do WSO2, este pode ser solicitado em qualquer dos métodos de obtenção de um access token, por indicação do scope *openid*.

A seguir apresentam-se os métodos de obtenção de *access token* que envolvem a interação de um utilizador com o browser (user agent):

Authorization Code (Code)

O método A*uthorization Code* (code) deverá ser usado em situações em que:

- É necessário criar um acesso de longa duração
- Há a necessidade de o utilizador interagir com um browser
- O token de acesso não deverá estar disponível no browser (e por consequência visível para o utilizador)

A obtenção access tokens é composta por dois passos. Num primeiro passo é solicitado, via browser, um código de autorização que deverá, num segundo passo (e por intermédio de uma chamada servidor – servidor), ser trocado por um access token (de curta duração). Este access token poderá então ser usado no consumo de recursos.

Na mensagem que contém o access token será ainda retornada a validade (em segundos) do access token gerado e um token de refrescamento (refresh token) que pode ser usado para obtenção de um novo access token (relacionado com gerado anteriormente) e desta forma prolongar a permissão de consumo recursos sem necessidade de executar a totalidade do processo.

Fluxo

1. Obtenção de um código de autorização (authorization code):

No pedido de obtenção de um authorization code é necessário especificar alguns parâmetros:

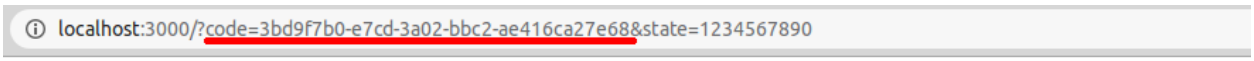
- URL Autorização: Parâmetro **URL Autorização**
- response_type: *code*
- client_id: Parâmetro **Client ID**
- state: Valor único usado pela aplicação na prevenção de ataques Cross-Site Request Forgery (CSRF)
- scope: Identificadores dos recursos a que a aplicação está a solicitar acesso
- redirect_uri: Parâmetro **Redirect URI**

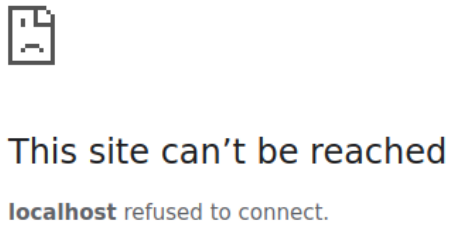
Exemplo:

curl --location --request GET 'https://wso2-gw.ua.pt/authorize?response_type=code&client_id=agh44RajMJcYvC&state=1234567890&scope=openid&redirect_uri=http://localhost:3000' --header 'Content-Type: application/x-www-form-urlencoded'

Neste momento, é apresentada ao utilizador a página de autenticação da UA para que este se possa autenticar.

Se a autenticação for feita com sucesso, o *authorization code* será devolvido no parâmetro *code* do URL para onde o utilizador é redirecionado:





2. Troca de authorization code por access token

No pedido de troca de um authorization code por um access token é necessário especificar alguns parâmetros:

- URL Token: Parâmetro **URL Token**
- Header Authorization: Basic <*base64 da string* ‘***Client ID***: ***Client Secret***’>
- grant_type: *authorization_code*
- code: Authorization code obtido no ponto 1.
- redirect_uri: Parâmetro **Redirect URI**

Exemplo:

curl --location --request POST 'https://wso2-gw.ua.pt/token?code=cf0e7e4c-d620-3f0a-ad6b-8d996ef65a44&redirect_uri=http://localhost&grant_type=authorization_code' --header 'Authorization: Basic dflgjsdhgsljkhsdfglkjsdhgijdfdsf' --header 'Content-Type: application/x-www-form-urlencoded'

{

"access_token": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",

"refresh_token": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",

"scope": "openid",

"id_token": "Loremipsumdolorsitamet3810ua.Loremipsumdolorsitamet3810ua consecteturadipiscingelit.Loremipsumdolorsitamet",

"token_type": "Bearer",

"expires_in": 3600

}

3. Consumo de APIs

No pedido de consumo de uma API é necessário incluir um header *Authorization* com o *access token* obtido no ponto 2.

Authorization: Bearer <access_token>

Exemplo:

curl --location --request GET 'https://wso2-gw.ua.pt/test_context/v1/health' \

--header 'Authorization: Bearer 9e1239ua-f0b4-3282-b784-74f589e075e0'

4. Prologamento das permissões de consumo por intermédio de um *refresh token*

No pedido de prologamento das permissões de consumo por intermédio de um refresh token é necessário especificar alguns parâmetros:

- URL Token: Parâmetro **URL Token**
- Header Authorization: Basic <*base64 da string '**Client ID**: **Client Secret**'*>
- grant_type: *refresh_token*
- refresh_token: Refresh token obtido no ponto 2.

Exemplo:

curl --location --request POST 'https://wso2-gw.ua.pt/token?refresh_token=389ua123-820e-325f-985f-d930b70679d9&grant_type=refresh_token' --header 'Authorization: Basic dlsdhfsdhfsdoghsdoiughsdofuihsdhfghdfhdhv' --header 'Content-Type: application/x-www-form-urlencoded'

Implicit

O método *Implicit* deverá ser usado em situações em que:

- Apenas é necessário acesso temporário a recursos
- O utilizador autentica-se com regularidade
- A aplicação corre num browser (usando JavaScript, Flash, etc.)
- O browser é altamente confiável e é improvável que o *access token* possa ficar visível para para utilizadores (ou aplicações) não confiáveis.

Usando o método *implicit*, o *access token* é devolvido imediatamente após um utilizador conceder acesso ao(s) recurso(s) solicitado(s), sem necessidade de nenhum passo adicional.

O método *implicit* não suporta a criação de tokens de refrescamento *(refresh token*). Quando o token de acesso *(access token*) expirar, a aplicação terá de executar o processo de autorização para obter novamente acesso aos recursos de que necessita.

Fluxo

1. Obtenção de um access token

No pedido de obtenção de um access token é necessário especificar alguns parâmetros:

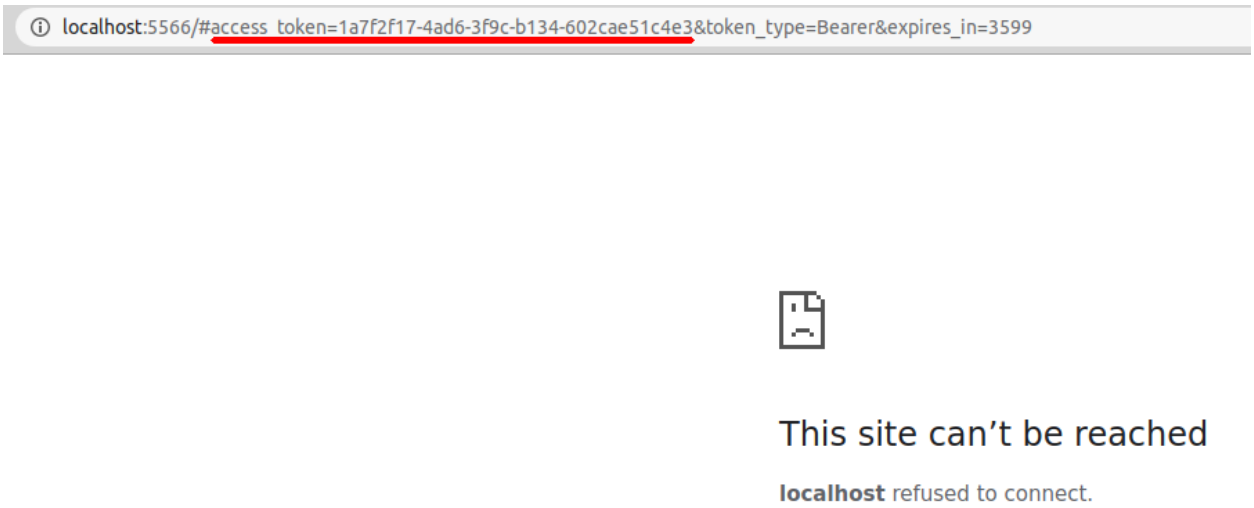
- URL Autorização: Parâmetro **URL Autorização**
- response_type: *token*
- client_id: Parâmetro **Client ID**
- state: Valor único usado pela aplicação na prevenção de ataques Cross-Site Request Forgery (CSRF)
- scope: Identificadores dos recursos a que a aplicação está a solicitar acesso
- redirect_uri: Parâmetro **Redirect URI**

Exemplo:

curl --location --request GET 'https://wso2-gw.ua.pt/authorize?response_type=token&client_id=agh44RajMJcYvC&state=1234567890&scope=openid&redirect_uri=http://localhost:5566' --header 'Content-Type: application/x-www-form-urlencoded'

Neste momento, é apresentada ao utilizador a página de autenticação da UA para que este se possa autenticar.

Se a autenticação for realizada com sucesso, o *acess token* será devolvido num fragmento do URL para onde o utilizador é redirecionado:



2. Consumo de APIs

No pedido de consumo de uma API é necessário incluir um header Authorization com o *access token* obtido no ponto 1.

Authorization: Bearer <access_token>

Exemplo:
curl --location --request GET 'https://wso2-gw.ua.pt/test_context/v1/health' --header 'Authorization: Bearer 1a7f2a12-4ad6-3f9c-b134-602uae51c4e3'

De notar que para ambos os métodos de obtenção de *access token*, os dados do utilizador devolvidos pelo IdP da UA podem ser consultados através do **URL Userinfo** por passagem do access token obtido.

Exemplo:
curl --location --request GET 'https://wso2-gw.ua.pt/userinfo' --header 'Authorization: Bearer 76e8ee11-d0b6-364d-8a44-0f0ua3ua1ac2'