SSO Specifications for OpenID Connect with Windows® Azure®

May 27, 2022

Introduction

OpenID Connect is a simple identity layer built on top of the OAuth 2.0 protocol. OAuth 2.0 defines mechanisms to obtain and use access tokens to access protected resources, but they do not define standard methods to provide identity information. OpenID Connect implements authentication as an extension to the OAuth 2.0 authorization process. It provides information about the end user in the form of an id_token that verifies the identity of the user and provides basic profile information about the user.

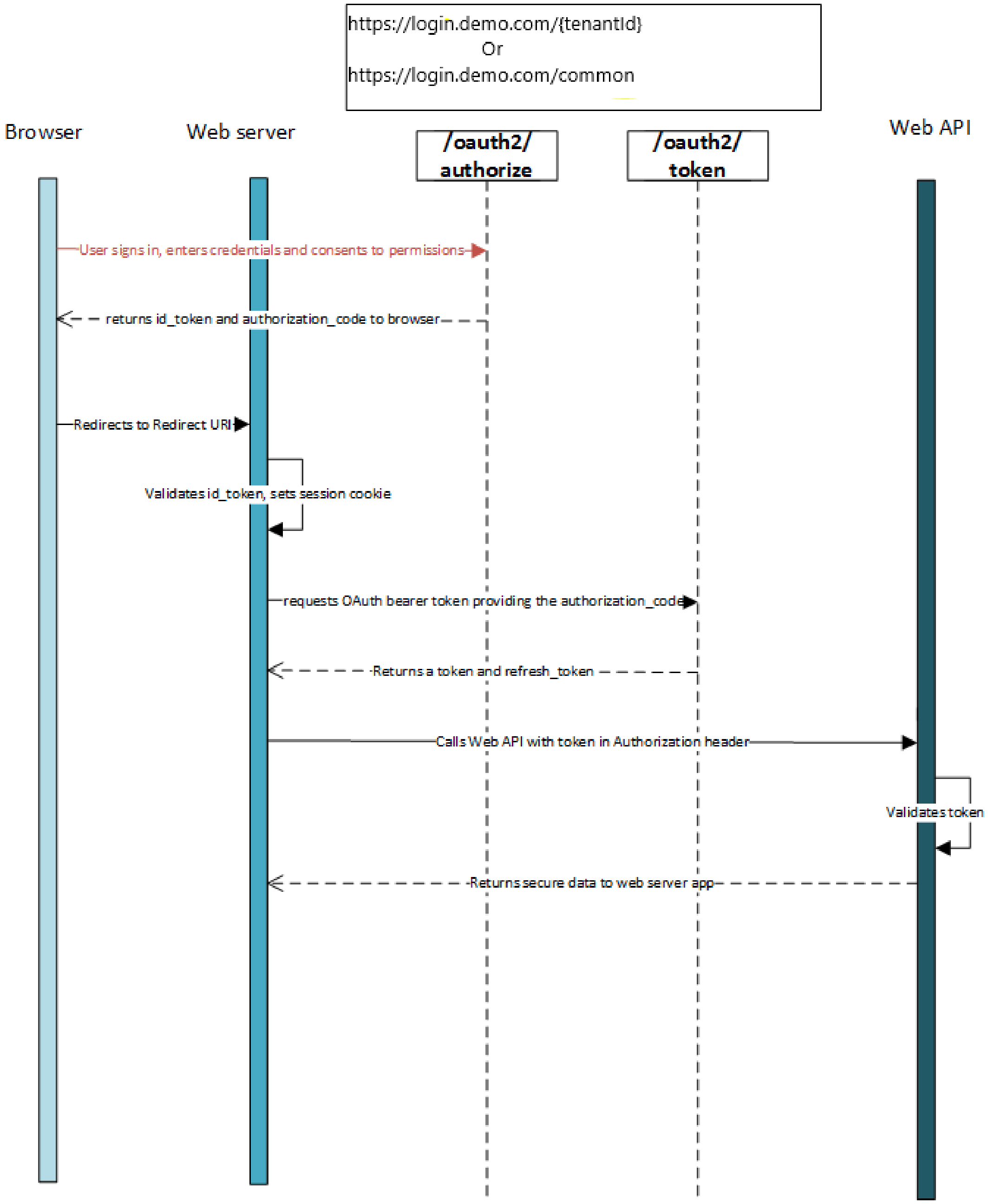
Register Application with AD Tenant

Register the application with Azure Active Directory (Azure AD) tenant. This will generate an Application ID for the application, as well as enable it to receive tokens.

1. Sign in to the Azure portal and select your Azure AD tenant.
2. In the left-hand navigation pane, click on Azure Active Directory.
3. Click **App Registrations** and click **New application registration**.
4. Follow the prompts and create a new application, which will involve providing the sign-on URL.
5. Once the registration is complete, Azure AD will assign the application a unique client identifier: the Application ID. The Application ID is required in the sign-in request. To find the application in the Azure portal, click **App registrations** and click **View all applications**.

The content in this document is the property of MealSuite and is intended for use to the addressed recipient(s) only. Distribution or disclosure of the content to other parties may violate copyrights, constitute trademark infringements and violate confidentiality.

Authentication Flow Using OpenID Connect


https://login.demo.com/{tenantId}
Or
https://login.demo.com/common
Browser
Web server
/oauth2/ authorize
/oauth2/ token
Web API
User signs in, enters credentials and consents to permissions
returns id_token and authorization_code to browser
Redirects to Redirect URI
Validates id_token, sets session cookie
requests OAuth bearer token providing the authorization_code
Returns a token and refresh_token
Calls Web API with token in Authorization header
Validates token
Returns secure data to web server app

Validating the id_token

After receiving the id_token, it is sent to a backend server and validation is performed there. Once validated, a session can begin with the user, using the claims in the id_token to obtain information about the user in the app.

Sign-In Request

When the web application needs to authenticate the user/browser, it directs the user/browser to the */authorize* endpoint.

Sample Request

A sample request is as follows:

```
GET https://login.demo.com/{tenant}/oauth2/authorize?
client_id=6731de76-14a6-49ae-97bc-6eba6914391e
&response_type=id_token
&redirect_uri=http%3A%2F%2Flocalhost%3a12345
&response_mode=form_post
&scope=openid
&state=12345
&nonce=7362CAEA-9CA5-4B43-9BA3-34D7C303EBA7
```

Parameter		Description
tenant	required	The {tenant} value in the path of the request is used to control who can sign into the application. The allowed values are tenant identifiers.
client_id	required	The Application ID assigned to the app when registered with Azure AD
response_type	required	Must include `id_token` for OpenID Connect sign-in.
scope	required	A space-separated list of scopes. For OpenID Connect, it must include the scope `openid`, which translates to the "Sign you in" permission in the consent UI.
nonce	required	A value included in the request, generated by the app, that is included in the resulting `id_token` as a claim.
redirect_uri	recommended	The redirect_uri of the app, where authentication responses are sent and received by the app.
response_mode	recommended	Specifies the method that should be used to send the resulting authorization_code back to the app.
state	recommended	A value included in the request that is returned in the token response.

Sample Response

A sample response, after the user/browser has been authenticated:

```
POST / HTTP/1.1
Host: localhost:12345
Content-Type: application/x-www-form-urlencoded

id_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Ik1uQ19WWmNB...&state=12345
```

Parameter	Description
id_token	The id_token that the app requested
state	A randomly generated unique value used for preventing cross-site request forgery attacks.

Error Response

Error responses are sent to the redirect_uri so the app can handle them appropriately.

```
POST / HTTP/1.1
Host: localhost:12345
Content-Type: application/x-www-form-urlencoded

error=access_denied&error_description=the+user+canceled+the+authentication
```

Parameter	Description
error	An error code string that can be used to classify types of errors that occur and can be used to react to errors. See table below for error code descriptions.
error_description	A specific error message that can help a developer identify the root cause of an authentication error.

Error Code	Description	Client Action
invalid_request	Protocol error, such as a missing required parameter.	Fix and resubmit the request. This is a development error and is typically caught during initial testing.
unauthorized_client	The client application is not permitted to request an authorization code.	This usually occurs when the client application is not registered in Azure AD or is not added to the user's Azure AD tenant.
access_denied	Resource owner denied consent	The client application can notify the user that it cannot proceed unless the user consents.

Error Code	Description	Client Action
unsupported_response_type	The authorization server does not support the response type in the request	Fix and resubmit the request. This is a development error and is typically caught during initial testing.
server_error	The server encountered an unexpected error.	Retry the request. These errors can result from temporary conditions. The client application might explain to the user that its response is delayed due to a temporary error.
temporarily_unavailable	The server is temporarily too busy to handle the request.	Retry the request. The client application might explain to the user that its response is delayed due to a temporary condition.
invalid_resource	The target resource is invalid because it does not exist, Azure AD cannot find it, or it is not correctly configured	This indicates the resource, if it exists, has not been configured in the tenant. The application can prompt the user with instruction for installing the application and adding it to Azure AD.