ADAL and Azure AD Graph

This guide provides setup requirements and steps to demonstrate how to use the Active Directory Authentication Library (ADAL) to obtain an Access Token from Azure AD. It also shows how to use the Azure AD Graph Client Library to programmatically create a user in Azure AD.

Contents

[Pre-Requisites 1](#_Toc430017095)

[Setup 1](#_Toc430017096)

[Demo Steps 1](#_Toc430017097)

[Clean Up 5](#_Toc430017098)

## Pre-Requisites

This section lists the pre-requisites required for this demonstration.

* An Azure subscription
* Visual Studio 2015
* Perform the setup requirements for **Demo 1 – App Integration** if you haven’t already.

## Setup

Estimated time: 5 minutes

|  |  |
| --- | --- |
| 1. Open Windows Explorer. 2. Copy the folder **.\Demos\4\_ADAL\_Graph** to **c:\azurecoe\demos\identity**. 3. Open Visual Studio. 4. Select **Tools > Code Snippets Manager**. 5. Set Language to **C#**. 6. Click on **My Code Snippets**. 7. Click **Import**. |  |
| 1. Select the file at **c:\azurecoe\demos\identity\4\_ADAL\_Graph\ADAL\_Demp.snippet**. 2. Click **Finish**.   In the Code Snippets Manager, expand the My Code Snippets folder and you should see ADAL\_Demo\_x\_x snippets as shown.   1. Click **OK**. |  |
| 1. Open the solution at **c:\azurecoe\demos\identity \4\_ADAL\_Graph\ADAL-Graph.Demo.Solution\ADAL-Graph.Demo.Solution.sln**. 2. Open **Program.cs** in the editor. 3. Rebuild the solution to pull down the NuGet packages. Note: This application doesn’t do anything yet. It is just a blank Console application with the ADAL and Azure AD Graph Client Library packages already added to the package library. That way you don’t have to take time hunting for them in the NuGet Package Manager.   **Stop here. This is where the demo steps will continue from**. |  |

## Demo Steps

Estimated time: 10 minutes

|  |  |
| --- | --- |
| 1. In the Azure Management Portal, go to the **APPLICATIONS** page of the **Azure CoE** Active Directory. 2. Click the **ADD** button at the bottom of the page. 3. Select **Add an application my organization is developing**. |  |
| 1. Set **Name** to **ADAL Graph Demo**. 2. Set **Type** to **Native Client Application**. 3. Click the right-arrow to continue. |  |
| 1. Set the **Redirect URI** to **http://adalgraphdemo**. 2. Click the checkmark button to create the application in Azure AD. 3. On the Quick Start page, expand the UPDATE YOUR CODE section to show the Redirect URI and the Client ID for the application. 4. Copy the Redirect URI to your clipboard. |  |
| 1. Switch over to Visual Studio. 2. Insert snippet **DAL\_Demo\_1\_Define\_Constants** under the designated comment. (**Ctrl-K,X**). 3. Copy the value in your clipboard to the **redirectUri** constant. 4. Switch over to the Azure Management Portal. 5. Copy the Client Id to your clipboard. 6. Copy the value in your clipboard to the **ClientId** constant. |  |
| 1. Switch over to the Azure Management Portal. 2. Go back to the **APPLICATIONS** page. 3. Click the **VIEW ENDPOINTS** button at the bottom of the page. 4. Explain that these are the endpoints that Azure AD provides for *your* active directory tenant. The GUID in the URL’s will be unique for each tenant. For this demo, you will be using the OAUTH 2.0 endpoints at the bottom. Later in this demo you will also use the Graph API endpoint. The GUID in the URL’s is the **tenant ID** that Azure AD assigned when your active directory was created. Copy this GUID to your clipboard.   Note: If you want to (and have time), open the Federation Metadata Document in separate browser window and show where the signing keys are, tenant id, and the endpoints supporting the other protocols such as WS-Federation and SAML-P.   1. Close the App Endpoints window. |  |
| 1. Click on the **ADAL Graph Demo** application. 2. Click on the **CONFIGURE** tab at the top of the page. 3. Under the permissions to other applications, add the permission for this application to read and write directory data.   Note: This is technically not necessary for this part of the demo. However, it will be needed in part-B of the demo and since it takes about 5 minutes for the permissions to take effect it’s a good idea to do it now.   1. Click **SAVE** at the bottom of the screen. |  |
| 1. Switch over to Visual Studio. 2. Copy the value in your clipboard to the **tenantId** constant. 3. While you are adding the tenant information, also set the **tenantName** constant to your tenant name (ie: azurecoe01).   Hint: If you don’t remember your tenant name, look at the username for John Doe in the USERS page in the Azure Management Portal. |  |
| 1. Insert snippet **DAL\_Demo\_2\_Add\_GetTokenForUser** under the designated comment. (**Ctrl-K,X**). 2. Explain this code while referencing back to the slide in the presentation talking about **AuthenticationContext, AcquireToken** and **AuthenticationResult**. 3. Insert snippet **DAL\_Demo\_3\_Call\_GetTokenForUser** under the designated comment. (**Ctrl-K,X**). |  |
| 1. Press **Ctrl-F5** to build and run the application. 2. When prompted, sign-in as **john@azurecoe01.onmicrosoft.com** with password **P@ssword1**. |  |
| 1. Show the access token in the console window. 2. Press enter to close the console window. |  |
| 1. Summarize: So far, you have shown how to use ADAL to    1. connect to Azure AD using the **AuthenticationContext** class,    2. authenticate a user and receive a token using the **AcquireToken** method,    3. View the access token that was returned in the **AuthenticationResult** instance. |  |

## Clean Up

To clean up after this demo perform the clean-up steps in part B of this demo.