

## Demo: Interactive authentication by using MSAL.NET

In this demo you'll learn how to perform the following actions:

- Use the `PublicClientApplicationBuilder` class in MSAL.NET
- Acquire a token interactively in a console application

### Prerequisites

This demo is performed in Visual Studio Code. We'll be using information from the app we registered in the *Register an app with the Microsoft Identity platform* demo.

### Login to the Azure Portal

1. Login to the portal: <https://portal.azure.com>
2. Navigate to the app registered in the previous demo.
  - Search for **App Registrations** in the top search bar of the portal.
  - Select the `az204appregdemo` app your registered earlier in this module.
  - Make sure you select **Overview** in the left navigation.

### Set up the console application

1. Open a PowerShell terminal.
2. Create a folder for the project and change in to the folder.

```
md az204-authdemo
cd az204-authdemo
```

3. Create the .NET console app.

```
dotnet new console
```

4. Open Visual Studio Code and open the `az204-authdemo` folder.

```
code .
```

### Build the console app

#### Add packages and using statements

1. Add the `Microsoft.Identity.Client` package to the project in a terminal in VS Code.

```
dotnet add package Microsoft.Identity.Client
```

2. Add using statements to include `Microsoft.Identity.Client` and to enable async operations.

```
using System.Threading.Tasks;
using Microsoft.Identity.Client;
```

3. Change the `Main` method to enable `async`.

```
public static async Task Main(string[] args)
```

## Add code for the interactive authentication

1. We'll need two variables to hold the Application (client) and Directory (client) IDs. You can copy those values from the portal. Add the code below and replace the string values with the appropriate values from the portal.

```
private const string _clientId = "APPLICATION_CLIENT_ID";
private const string _tenantId = "DIRECTORY_TENANT_ID";
```

2. Use the `PublicClientApplicationBuilder` class to build out the authorization context.

```
var app = PublicClientApplicationBuilder
    .Create(_clientId)
    .WithAuthority(AzureCloudInstance.AzurePublic, _tenantId)
    .WithRedirectUri("http://localhost")
    .Build();
```

Code	Description
<code>.Create</code>	Creates a <code>PublicClientApplicationBuilder</code> from a <code>clientId</code> .
<code>.WithAuthority</code>	Adds a known Authority corresponding to an ADFS server. In the code we're specifying the Public cloud, and using the tenant for the app we registered.

## Acquire a token

When you registered the `az204appregdemo` app it automatically generated an API permission `user.read` for Microsoft Graph. We'll use that permission to acquire a token.

1. Set the permission scope for the token request. Add the following code below the `PublicClientApplicationBuilder`.

```
string[] scopes = { "user.read" };
```

2. Request the token and write the result out to the console.

```
AuthenticationResult result = await app.AcquireTokenInteractive(scopes).ExecuteAsync();

Console.WriteLine($"Token:\t{result.AccessToken}");
```

## Run the application

1. In the VS Code terminal run `dotnet build` to check for errors, then `dotnet run` to run the app.
2. The app will open the default browser prompting you to select the account you want to authenticate with. If there are multiple accounts listed select the one associated with the tenant used in the app.

- If this is the first time you've authenticated to the registered app you will receive a **Permissions requested** notification asking you to approve the app to read data associated with your account. Select **Accept**.



## Permissions requested

[App info](#)

This app would like to:

- ☒ Maintain access to data you have given it access to
- ☒ Sign you in and read your profile
- ☐ Consent on behalf of your organization

Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. You can change these permissions at <https://myapps.microsoft.com>. [Show details](#)

Cancel

Accept

- You should see the following results in the console.

Token: eyJ0eXAiOiJKV1QiLCJub25jZSI6I1VhU.....

✓ **Note:** We'll be reusing this project in the *Retrieving profile information by using the Microsoft Graph SDK* demo in the next lesson.