# Configuring Azure AD Applications for Power BI Embedding

**Setup Time**: 60 minutes

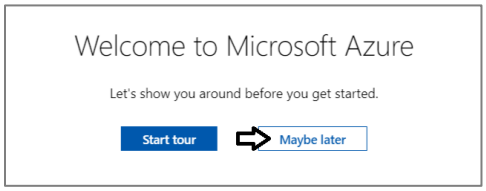
**Lab Folder**: C:\Student\Modules\02_AzureActiveDirectory\Lab

**Overview**: In this lab, you will upload PBIX files into your personal workspace and you can begin designing dashboard and reports in the browser. The lab will also step you through downloading and installing Power BI Desktop as well as publishing a Power BI Desktop project to the Power BI service. You create a new app workspace and populate it with a dataset, a report and a dashboard. In the final exercises, you will program against the Power BI Service API.

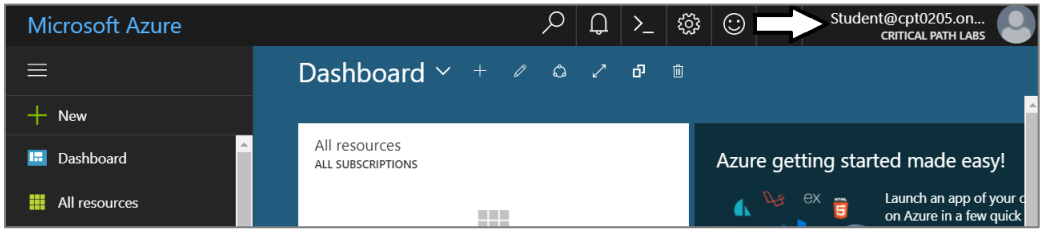
## Exercise 1: Register a Native Client Application using the Azure Portal

In this exercise, you will register a new application with Azure AD and you will configure the application's required permissions to access the Power BI Service API.

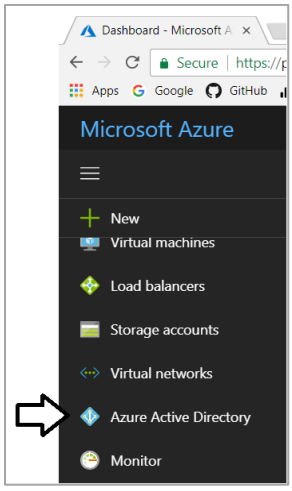
1. Log into the Azure Portal
   a) In the browser, navigate to the Azure portal at https://portal.azure.com.
   b) When you are prompted to log in, provide the credentials to log in with your Office 365 user account name.
   c) If you are prompted to start a tour of Microsoft Azure, click **Maybe later**.



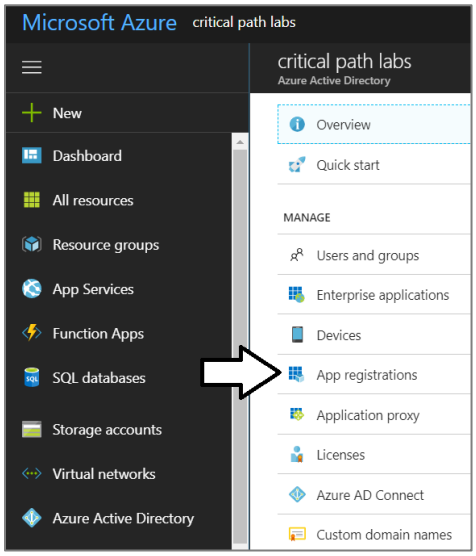

   d) Once you are log into the Azure portal, check the email address in the login menu in the upper right to make sure you are logged in the Azure portal with the correct identity.



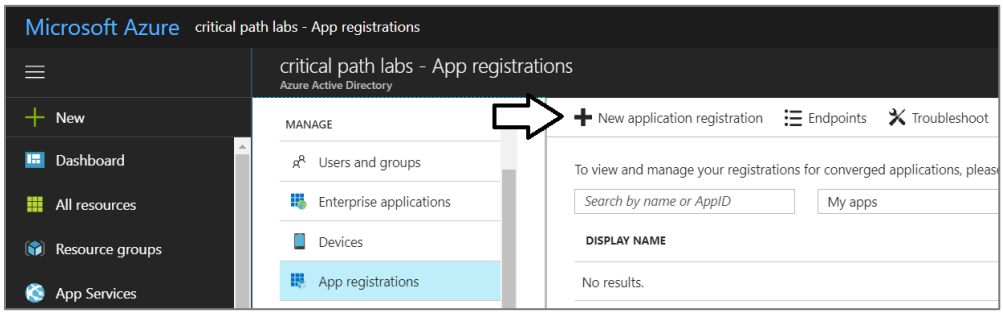

2. Register a new Azure application.
   a) In the left navigation, scroll down and click on the link for **Azure Active Directory**.



b) Click the link for **App registration**.




c) Click **New application registration**.



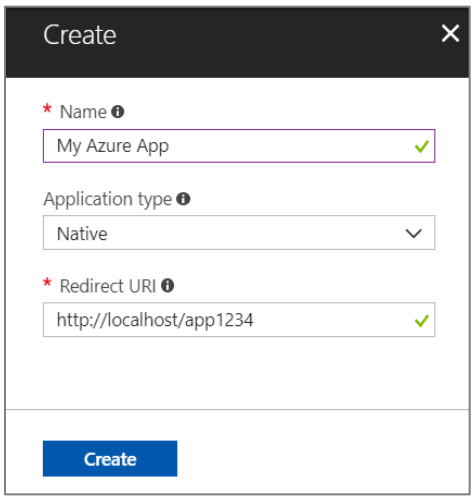

d) In the Create dialog…

i) Add a **Name** of **My Azure App**.

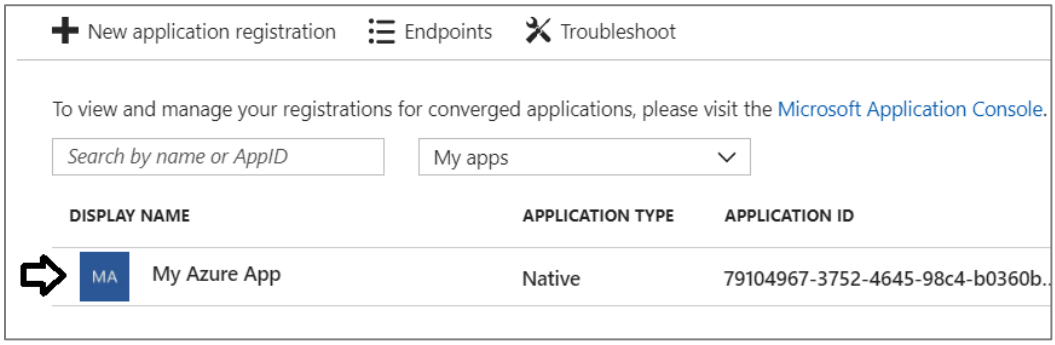
ii) Set the **Application type** to **Native**.

iii) Set the **Redirect URI** to http://localhost/app1234.

iv) Click the **Create** button to create the new application.

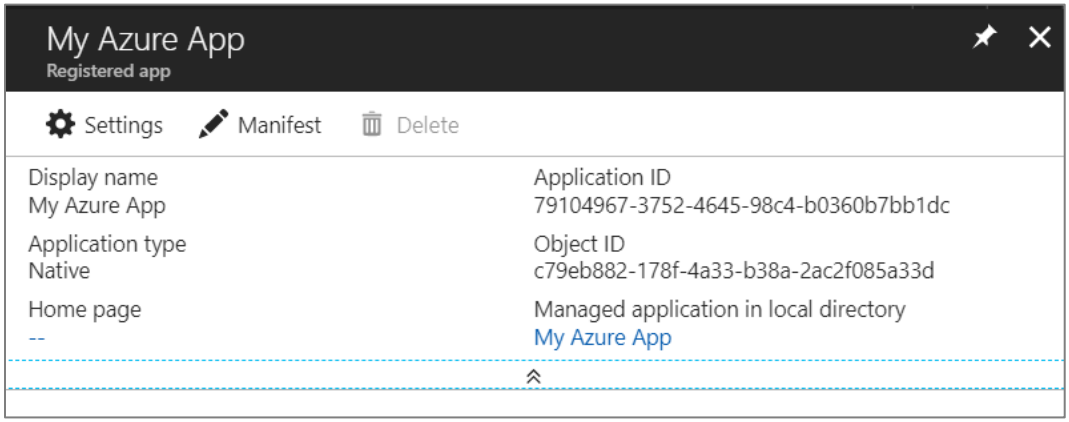



e) Once should now see the new application.



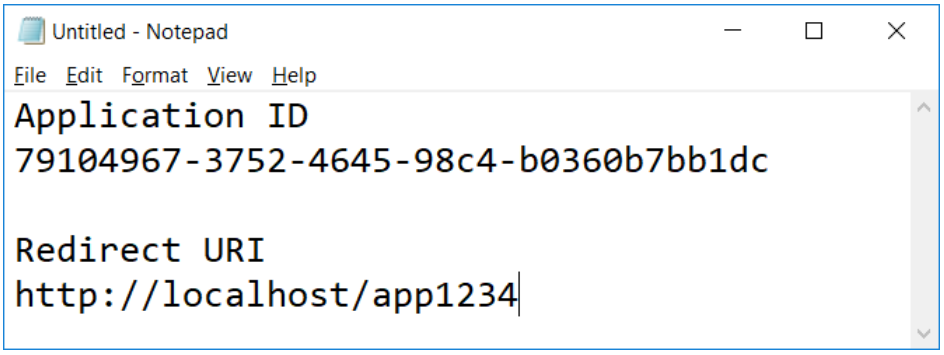

3. Copy the GUID for the Application ID.

a) Click on the link for the new application named **My Azure App** to get to the details page.

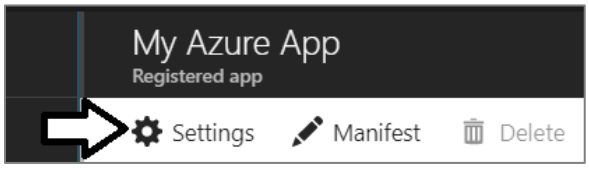



b) Copy the Application ID to the Windows clipboard.

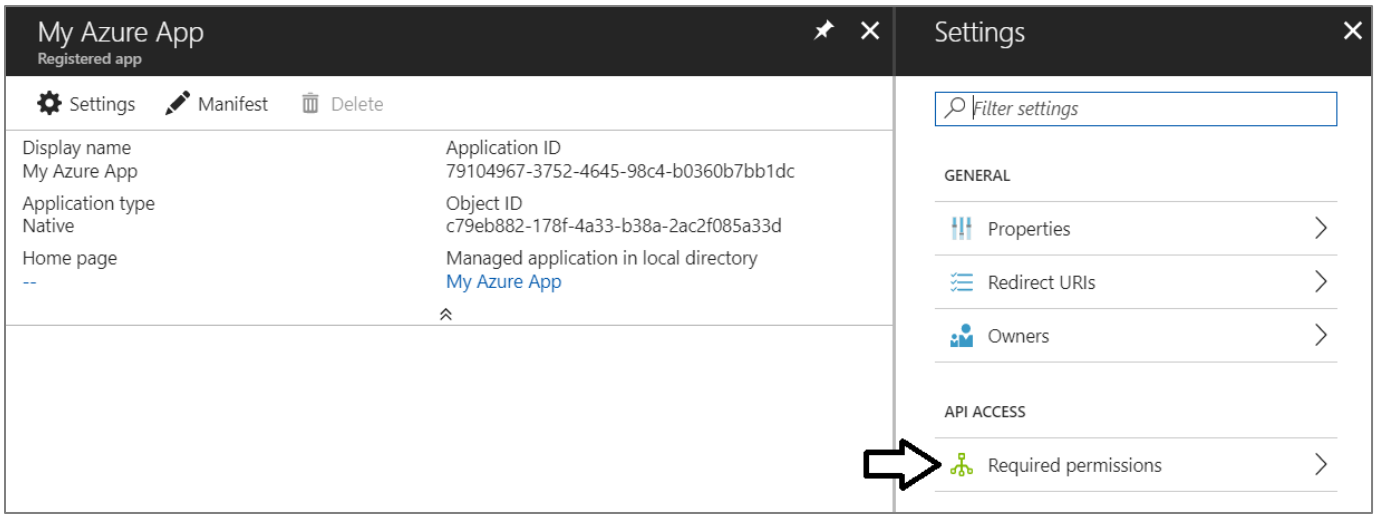

c) Launch Notepad and paste the Application ID into a new document. Also add the value of the Redirect URI.



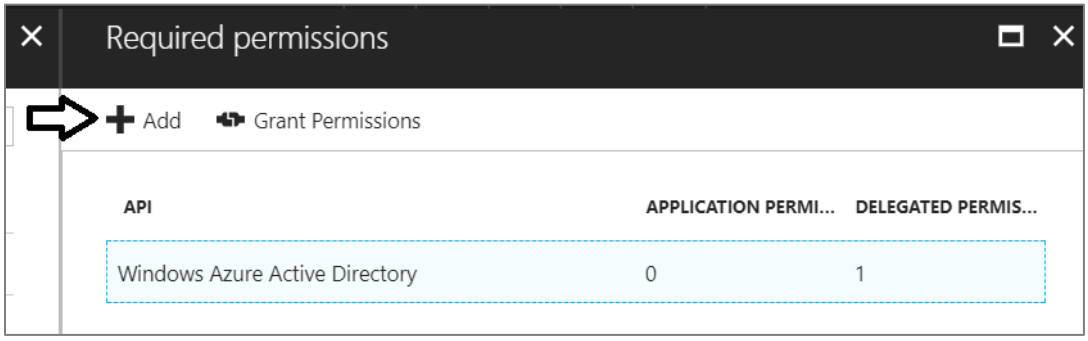

d) Click on the **Settings** link to configure application settings,



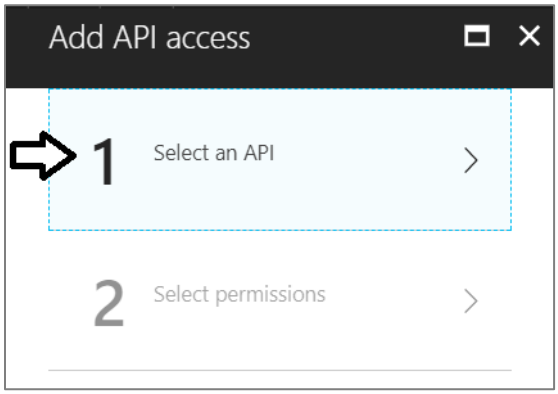

e) Click **Required permissions**.



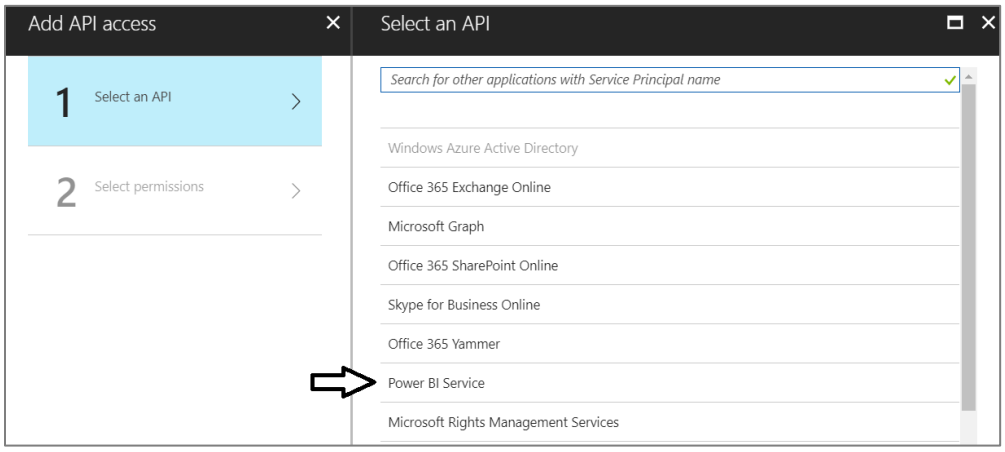

f) Click the **Add** button on the **Required permissions** blade.




g) Click the **Select an API** option in the **Add API** access blade.

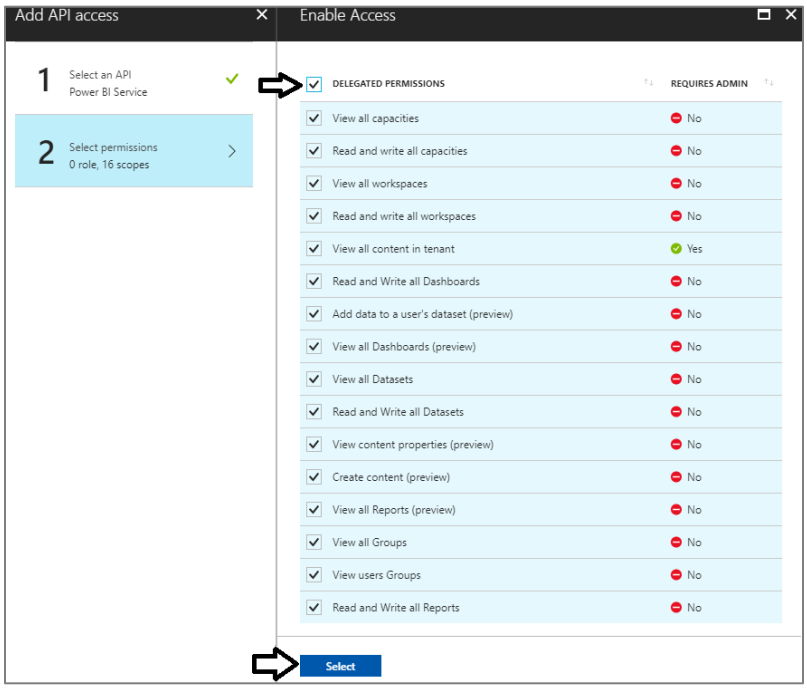



h) In the **Select an API** blade, click **Power BI Service**.

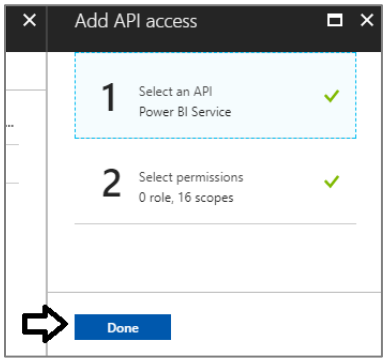



i) In the **Enable Access** blade, click the top checkbox for **DELEGATED PERMISSIONS** to select all the permissions.

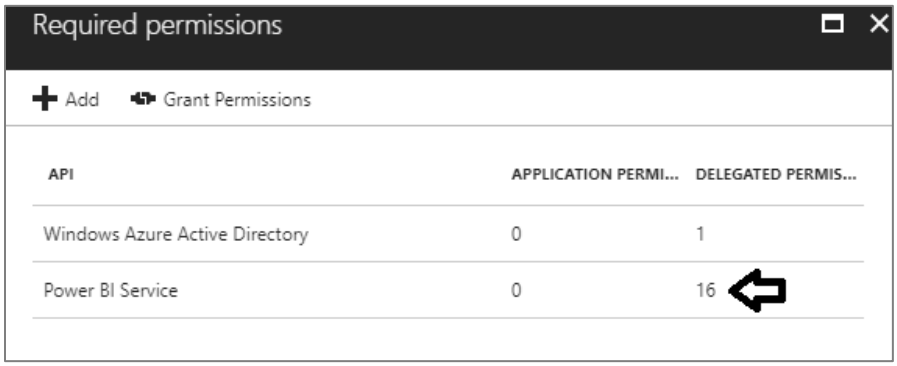
j) Once you have selected all the permissions, click the **Select** button at the bottom of the blade.




k) Click the **Done** button at the bottom of the **Add API Access** blade.




l) At this point, you should be able to verify that the Power BI Service has been added to the **Required permissions** list.

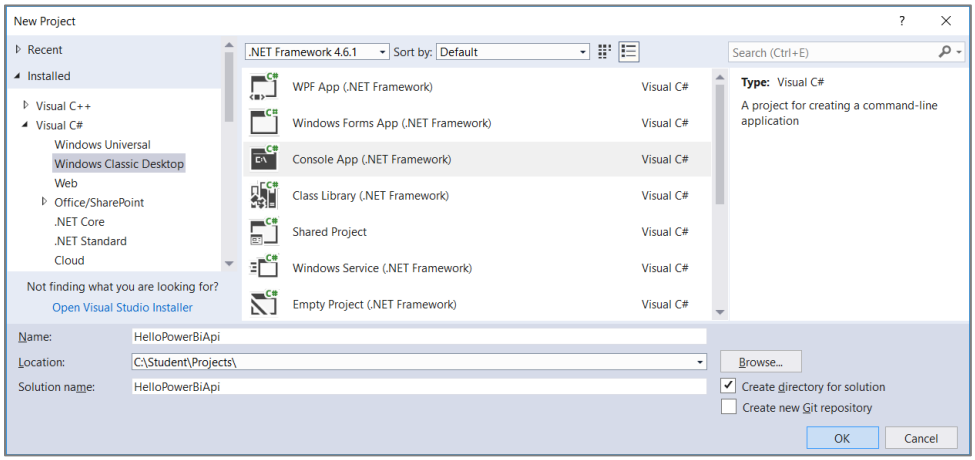



You are now done registering your application with Azure AD.

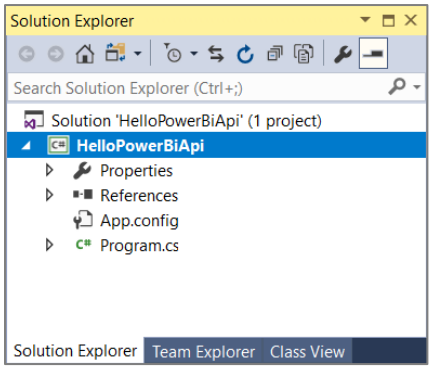
## Exercise 2: Call the Power BI Service API from a C# Console Application

In this exercise, you will create a simple C# Console application to call into the Power BI Service API.

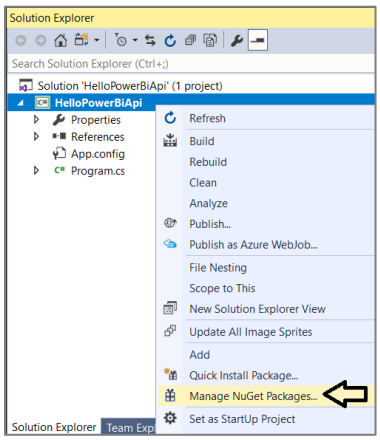
1. Create a new C# Console application in Visual Studio.
   a) Launch Visual Studio.
   b) Create a new project by running the **File > New Project** command.
   c) Select a project type of Console App from the Visual C# project templates.
   d) Give the project a name of **HelloPowerBiApi** and click OK.




   e) You should now have a new project named **HelloPowerBiApi**.

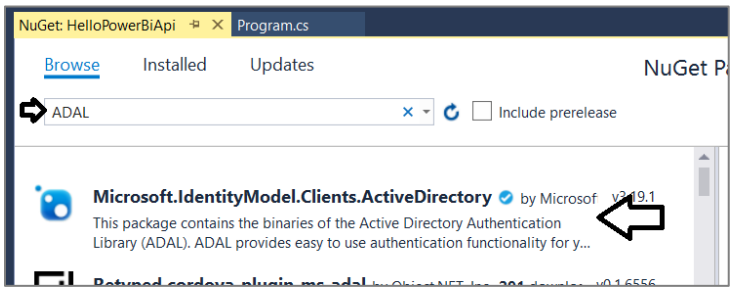



2. Add NuGet package to the project.
   a) Right-click the top-level node for the **HelloPowerBiApi** project and select **Manage NuGet Packages…**.

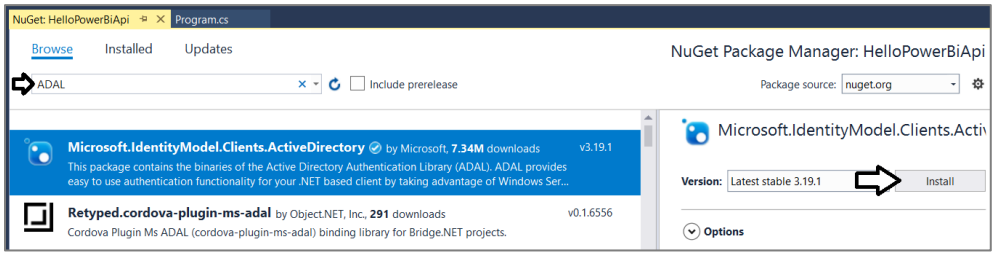



b) Click the Browse tab and type ADAL into the search box.

c) Locate the package **Microsoft.IdentityModel.Clients.ActiveDirectory**. This is the Active Directly Authentication library.

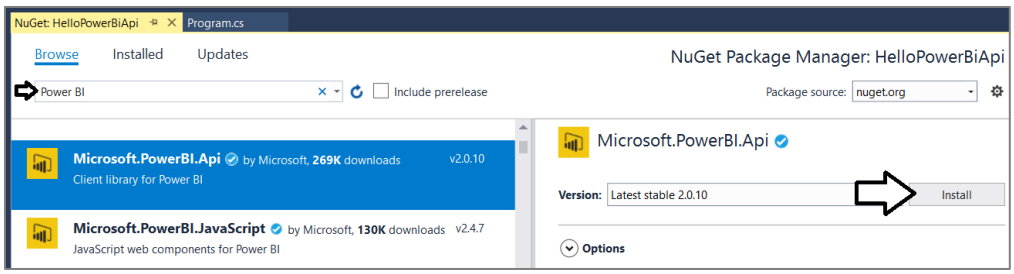



d) Select and install **Microsoft.IdentityModel.Clients.ActiveDirectory**.




e) When prompted about the licensing agreement, click **I Agree**.

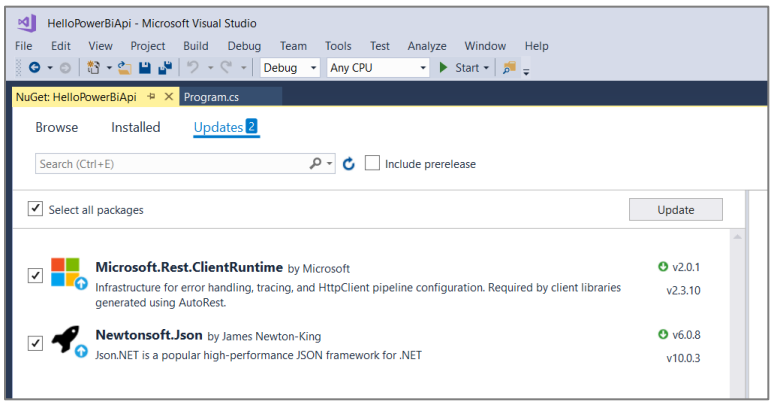
f) Search for Power BI and then find and install the **Microsoft.PowerBI.Api**.




g) When prompted about the licensing agreement, click **I Agree**.

3. Update all NuGet packages.

a) Navigate to the **Update** tab and update any packages that have updates available.




b) Close the window for the Nuget Package Manager.

4. Add code to the project

a) Using Windows Explorer, locate the file named **PowerBiStarter.cs.txt** located in the Student folder at the following path.

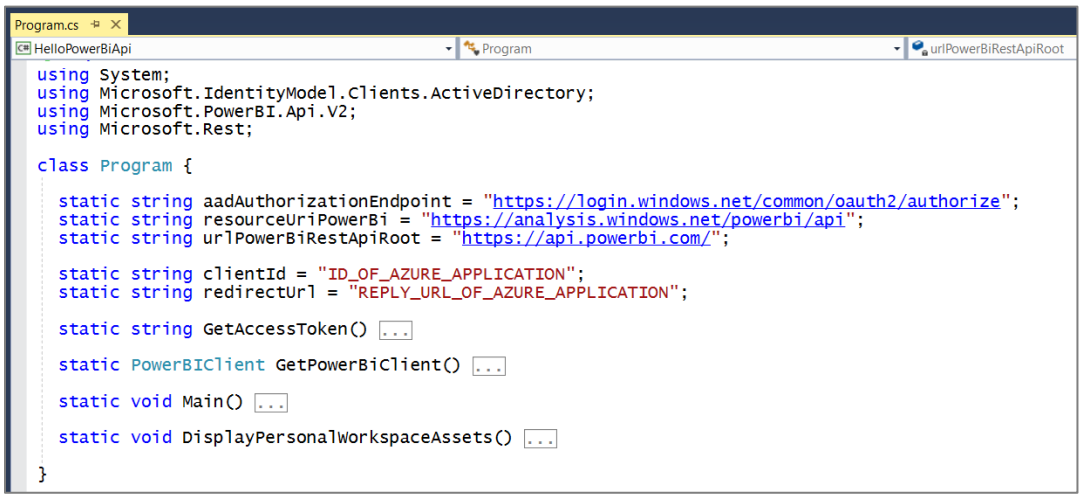
b) Open the file named **PowerBiStarter.cs.txt** in Notepad and copy its contents into the Window clipboard.

c) Return to the **HelloPowerBiApi** project in Visual Studio.

d) Open the source file named **program.cs**.

e) Delete all the code inside **program.cs** and replace it with the content you copied into the Windows clipboard.

f) You should now have the basic code for a simple application which access the Power BI Service API.




5. Update the code with your Application ID and Redirect URI.

a) Locate the section of the code with the static properties named **clientId** and **redirectUrl**.

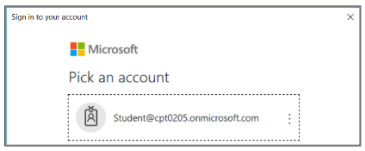
```
static string clientId = "ID_OF_AZURE_APPLICATION";
static string redirectUrl = "REPLY_URL_OF_AZURE_APPLICATION";
```

b) Replace these values with the values you copied into Notepad earlier.

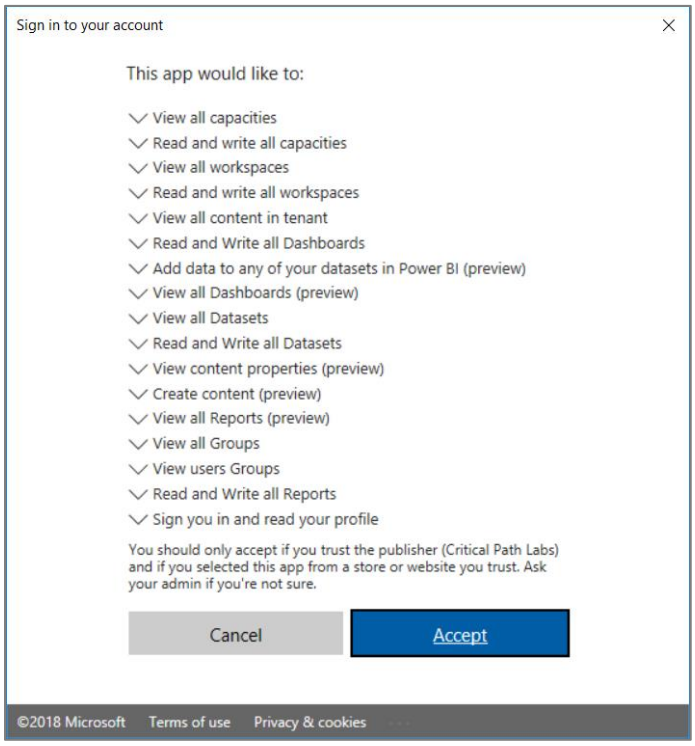
```
static string clientId = "79104967-3752-4645-98c4-b0360b7bb1dc";
static string redirectUrl = "http://localhost/app1234";
```

c) Save your changes to **program.cs**.

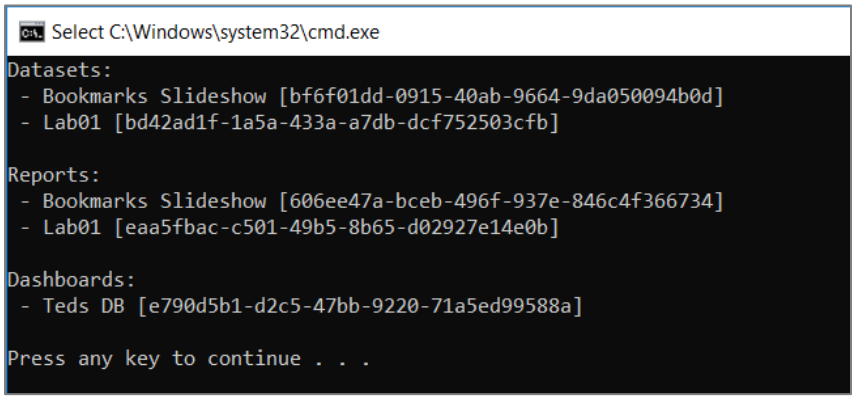
6. Run the application to call to the Power BI Service API.
   a) Press the {F5} key to begin a debugging session.
   b) When promoted to sign in, log in using your Power BI account and credentials.




   c) When prompted with the following screen, click **Accept**.




   d) The application should call into the Power BI Service API and retrieve data about the contents of your personal workspace.




Congratulations. You have now successfully called into the Power BI Service API.

## Exercise 3: Configure an Azure AD Application for Third Party Embedding

In this exercise, you will create a new app workspace and then you will work to populate this workspace with embeddable content which will include a dataset, a report and a dashboard.

7. Make sure you are logged into the Power BI service with your primary Office 365 user account.

## Exercise 4: Configure an Azure AD Application for First Party Embedding

In this exercise, you will create a new app workspace and then you will work to populate this workspace with embeddable content which will include a dataset, a report and a dashboard.

8. Make sure you are logged into the Power BI service with your primary Office 365 user account.

## Exercise 5: Configure an Azure AD Application as an SPA with Implicit Flow

In this exercise, you will create a new app workspace and then you will work to populate this workspace with embeddable content which will include a dataset, a report and a dashboard.

9. Make sure you are logged into the Power BI service with your primary Office 365 user account.