Task 1: Open the Azure portal

1. On the taskbar, select the **Microsoft Edge** icon.
2. In the open browser window, browse to the Azure portal at https://portal.azure.com, and then sign in with the account you'll use for this lab.

**Note**: If this is your first time signing in to the Azure portal, you'll be offered a tour of the portal. Select **Get Started** to skip the tour and begin using the portal.

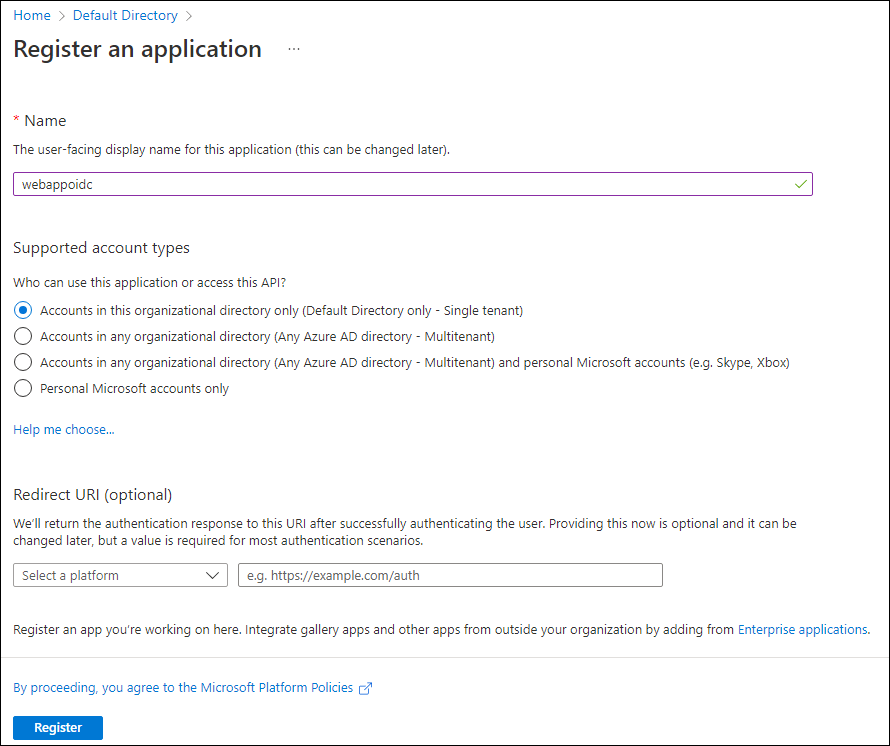
Task 2: Register an application in Entra ID Directory

1. In the Azure portal, use the **Search resources, services, and docs** text box to search for **Entra ID** and, in the list of results, select **Microsoft Entra ID**.

**Note**: This redirects your browser session to the blade of the Microsoft Entra ID tenant associated with your Azure subscription.

1. On the **Microsoft Entra ID** blade, select **App registrations** in the **Manage** section.
2. In the **App registrations** section, select **+ New registration**.
3. In the **Register an application** section, perform the following actions, and then select **Register**:

| **Setting** | **Action** |
| --- | --- |
| **Name** text box | enter webappoidc |
| **Supported account types** list | Select **Accounts in this organizational directory only (Default Directory only - Single tenant)** |

1. **Note**: The name of the tenant might differ depending on your Azure subscription.
2. The following screenshot displays the configured settings in the **Register an application** section.
3. 

Task 3: Record unique identifiers

1. On the **webappoidc** application registration blade, select **Overview**.
2. In the **Overview** section, find and record the value of the **Application (client) ID** text box. You'll use this value later in the lab.
3. In the **Overview** section, find and record the value of the **Directory (tenant) ID** text box. You'll use this value later in the lab.

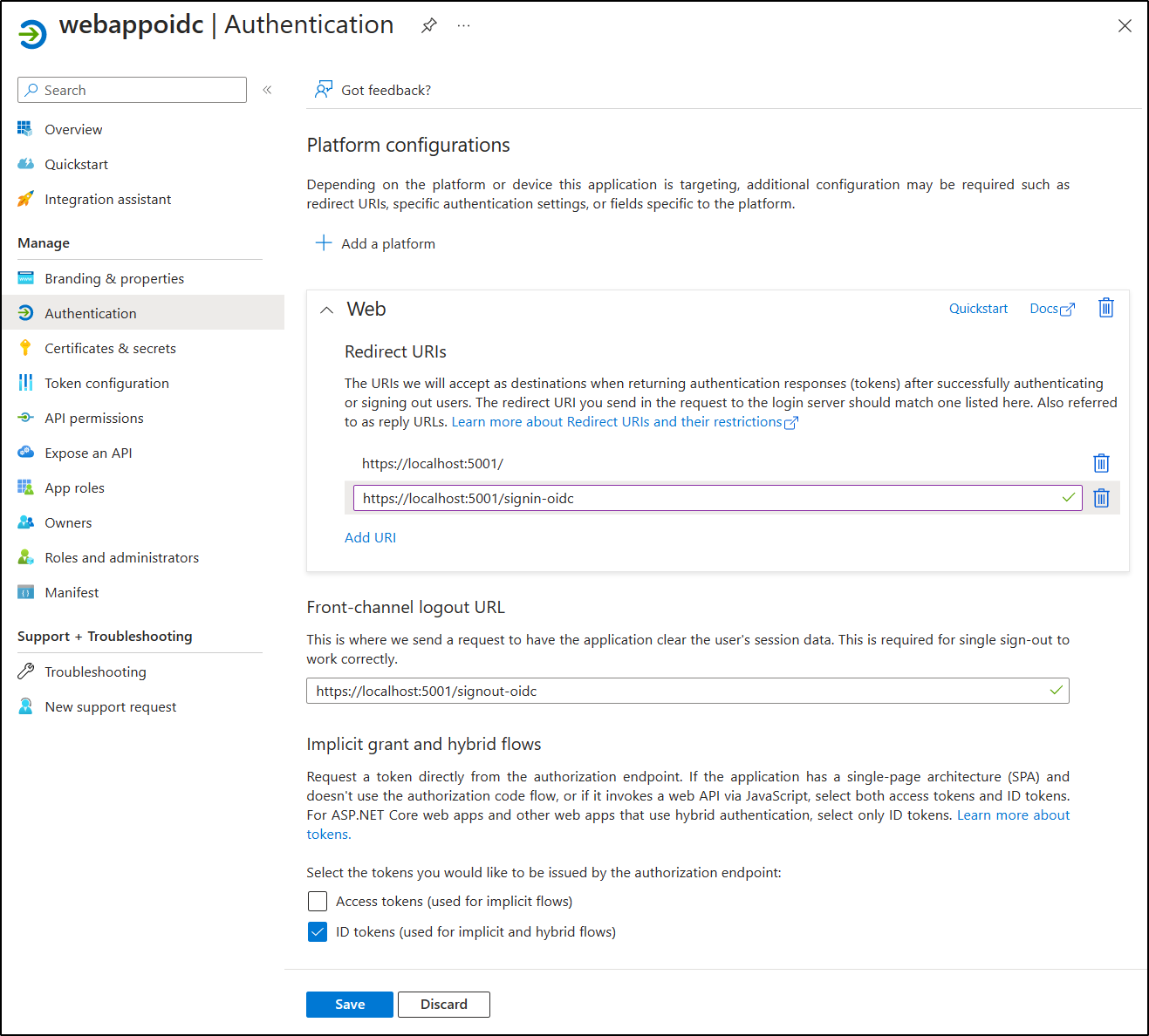
Task 4: Configure the application authentication settings

1. On the **webappoidc** application registration blade, select **Authentication** in the **Manage** section.
2. In the **Authentication** section, perform the following actions, and select **Configure**:

| **Setting** | **Action** |
| --- | --- |
| **Platform configurations** section | Select **+ Add a platform** |
| **Configure platforms** blade | Select **Web** |
| **Redirect URIs** text box | Enter https://localhost:5001/ |
| **Front-channel logout URL** text box | Enter https://localhost:5001/signout-oidc |

1. Back in the **Platform configurations** section, select **Add URI**, and then enter https://localhost:5001/signin-oidc.
2. In the **Implicit grant and hybrid flows** section, select **ID tokens (used for implicit and hybrid flows)**.
3. Select **Save**.

The following screenshot displays the configured settings on the **Authentication** blade.



Task 5: Create a Microsoft Entra ID user

1. In the Azure portal, select the **Cloud Shell** icon Cloud Shell icon to open a the Cloud Shell. If Cloud Shell defaults to a Bash session, select **Switch to PowerShell** in the **Cloud Shell** menu and then select **Confirm**.

**Note**: The **Cloud Shell** icon is represented by a greater than sign (>) and underscore character (\_).

If this is the first time you're starting **Cloud Shell** you will be presented with a **Getting started** prompt. Select **No storage account required**, then select **Apply**.

1. In the **Cloud Shell** pane, run the following command to sign in to the Microsoft Entra tenant associated with your Azure subscription:

powershell

Connect-AzureAD

1. Run the following command to retrieve and display the primary Domain Name System (DNS) domain name of the Microsoft Entra tenant:

powershell

$aadDomainName = ((Get-AzureAdTenantDetail).VerifiedDomains)[0].Name

$aadDomainName

**Note**: Record the value of the DNS domain name. You'll use this value later in the lab.

1. Run the following commands to create Microsoft Entra ID users that you'll use to test Microsoft Entra ID authentication:

Please skip this step as the user account has already been created for you.

powershell

$passwordProfile = New-Object -TypeName Microsoft.Open.AzureAD.Model.PasswordProfile

$passwordProfile.Password = 'Pa55w.rd1234'

$passwordProfile.ForceChangePasswordNextLogin = $false

New-AzureADUser -AccountEnabled $true -DisplayName 'aad\_lab\_user-52410134@LODSPRODMSLEARNMCA.onmicrosoft.com' -PasswordProfile $passwordProfile -MailNickName 'aad\_lab\_user-52410134@LODSPRODMSLEARNMCA.onmicrosoft.com' -UserPrincipalName "aad\_lab\_user-52410134@LODSPRODMSLEARNMCA.onmicrosoft.com@$aadDomainName"

1. Run the following command to identify the user principal name (UPN) of the newly created Microsoft Entra ID user:

powershell

(Get-AzureADUser -Filter "DisplayName eq 'aad\_lab\_user-52410134'").UserPrincipalName

**Note**: Record the UPN. You'll use this value later in the lab.

1. Close the Cloud Shell pane.

Review

In this exercise, you registered a single-tenant Azure AD application and created an Azure AD user account.

Exercise 2: Create a single-tenant ASP.NET web app

Task 1: Create an ASP.NET web app project

1. On the lab computer, start **Command Prompt**.
2. From the command prompt, run the following commands to create and set the current directory to **Allfiles C:\Allfiles\Labs\06\Starter\OIDCClient**:

cmd

C:

cd C:\Allfiles\Labs\06\Starter\OIDCClient

1. Run the following commands to create a new .NET Core web app based on the Model View Controller (MVC) template (replace the placeholders <application\_ID>, <tenant\_ID>, and <domain\_Name> with the corresponding values you recorded earlier in this lab):

cmd

dotnet new mvc --auth SingleOrg --client-id <application\_ID> --tenant-id <tenant\_ID> --domain <domain\_Name>

rmdir .\obj /S /Q

**Note**: If you received an error in the terminal, it is possible you're using PowerShell. It that case remove the /S /Q flags and rerun the command.

1. Run the following command to start Visual Studio Code.

cmd

code .

If you have been prompted **Do you trust the authors of all files in this folder?**, select **Yes, I trust the authors**.

1. In the Visual Studio Code **Explorer** pane, review the autogenerated folder structure that represents an MVC web app.
2. Navigate to the **Properties** folder, open the **launchSettings.json** file, and then apply the following changes:

| **Section** | **Property** | **Value** |
| --- | --- | --- |
| **iisSettings** | **sslPort** | **44321** |
| **https** | **applicationUrl** | https://localhost:5001 |

1. **Note**: The port numbers must match the value you specified when creating the Microsoft Entra ID app registration.
2. Save and close the file.
3. In the Visual Studio Code **Explorer** pane, select **OIDCClient.csproj**.
4. Ensure that the value of <TargetFramework> element is set to **net8.0**.
5. Ensure that version of the Microsoft.AspNetCore.Authentication.JwtBearer and Microsoft.AspNetCore.Authentication.OpenIdConnect NuGet packages is set to **8.0.0**.
6. Verify that the content of the **OIDCClient.csproj** file resembles the following listing (the value of the UserSecretsIdwill differ) and save the changes.

csharp

<Project Sdk="Microsoft.NET.Sdk.Web">

<PropertyGroup>

<TargetFramework>net8.0</TargetFramework>

<Nullable>enable</Nullable>

<ImplicitUsings>enable</ImplicitUsings>

<UserSecretsId>aspnet-OIDCClient-5249f5b2-6266-4a13-b8b4-e79c6443aabc</UserSecretsId>

</PropertyGroup>

<ItemGroup>

<PackageReference Include="Microsoft.AspNetCore.Authentication.JwtBearer" Version="8.0.0-rc.2.23480.2" NoWarn="NU1605" />

<PackageReference Include="Microsoft.AspNetCore.Authentication.OpenIdConnect" Version="8.0.0-rc.2.23480.2" NoWarn="NU1605" />

<PackageReference Include="Microsoft.Identity.Web" Version="2.13.0" />

<PackageReference Include="Microsoft.Identity.Web.UI" Version="2.13.0" />

<PackageReference Include="Microsoft.Identity.Web.DownstreamApi" Version="2.13.0" />

</ItemGroup>

</Project>

1. Close the **OIDCClient.csproj** file.
2. Navigate to the **Views\Shared** folder, and then open the **\_LoginPartial.cshtml** file.
3. Verify that the asp-area attribute in each span element references MicrosoftIdentity, as in the following line:

csharp

<a class="nav-link text-dark" asp-area="MicrosoftIdentity" asp-controller="Account" asp-action="SignOut">Sign out</a>

1. Close the file without making any changes.
2. Open the file **appsettings.json** in the root folder and review the content of the **AzureAd** object, including the following elements:

| **Element** | **Value** |
| --- | --- |
| Instance | https://login.microsoftonline.com/ |
| Domain | Primary DNS domain of the Azure AD tenant associated with your Azure subscription |
| TenantId | GUID of the Azure AD tenant |
| ClientId | Application (client) ID of the application you registered in the Azure AD tenant |
| CallbackPath | /signin-oidc |

1. Close the file without making any changes.
2. In the Visual Studio Code **Explorer** pane, select **Program.cs**.
3. Verify that the file contains the following **using** directives:

csharp

using Microsoft.AspNetCore.Authentication.OpenIdConnect;

using Microsoft.Identity.Web;

using Microsoft.Identity.Web.UI;

1. Verify that the file contains the following lines that add the relevant authentication services to the container:

csharp

// Add services to the container.

builder.Services.AddAuthentication(OpenIdConnectDefaults.AuthenticationScheme)

.AddMicrosoftIdentityWebApp(builder.Configuration.GetSection("AzureAd"));

1. Verify that the file contains the following lines that add a controller and Razor pages for the accounts management:

csharp

builder.Services.AddRazorPages()

.AddMicrosoftIdentityUI();

1. Save and close the file.

Task 2: Test the single-tenant web app in a single-tenant scenario

1. In the **Visual Studio Code** window, from its top menu bar, go to **Terminal** menu and select **New Terminal**.
2. In the **Terminal** panel, the current path of current working directory should be **C:\Allfiles\Labs\06\Starter\OIDCClient**. Run the following commands to build the .NET web app:

dotnet build

**Note**: If there are any build errors, review the files in the **Allfiles C:\Allfiles\Labs\06\Solution\OIDCClient** folder. Ignore any warning messages.

1. Run the following command to generate a self-signed certificate and configure the local computer to trust it:

dotnet dev-certs https --trust

1. If prompted to install the autogenerated certificate, select **Yes**.
2. From the terminal prompt, run the following command to run the .NET web app:

dotnet run

1. Start the Microsoft Edge browser in the **InPrivate** mode, and then navigate to the https://localhost:5001 URL.
2. If presented with the **Your connection isn't private** message, select **Advanced**, and then select the **Continue to localhost (unsafe)** link.
3. In the open browser window, when prompted, authenticate by using the UPN aad\_lab\_user-52410134@LODSPRODMSLEARNMCA.onmicrosoft.com Azure AD account you created earlier in this lab with HBk90!d!eDK@ as its password.

**Note**: If you are prompted with a **Help us protect your account** window, select **Skip for now**.

**Note**: If you are presented with the message **Need admin approval**, then you will need to use a Microsoft Entra ID with elevated privileges instead to complete the remainder of this task. To do so, select the link **Have an admin account? Sign in with that account**, sign in with an admin account, and proceed to the next step.

1. The browser window will automatically open the **Permissions requested** webpage.
2. Review the requested permissions, which include **View your basic profile** and **Maintain access to data you have given it access to**.
3. Select **Accept**.
4. Review the **Welcome** home page of the target site displayed by the browser and verify that the UPN of the **aad\_lab\_user-52410134@LODSPRODMSLEARNMCA.onmicrosoft.com** Microsoft Entra ID account appears in the browser window.
5. On the **Welcome** page, select **Sign out**.
6. When prompted to select the account to sign out, select the **aad\_lab\_user-52410134@LODSPRODMSLEARNMCA.onmicrosoft.com** Microsoft Entra ID account. You'll be automatically redirected to the **Signed out** page.
7. Close the Microsoft Edge browser.