

Call Web APIs from Data Access Layer in .NET MAUI Labs

Perform these labs on your own computer using Visual Studio 2022 or later to ensure you understand the lessons presented in the corresponding videos and lectures.

Lab 1: Add Data Layer API Project to Call Minimal Web API

Add a new **Class Library** project to your solution and set the name to **AdventureWorks.DataLayer.API**.

Delete the **Class1.cs** file.

Right mouse-click on the **Dependencies** folder and add project references to

- AdventureWorks.EntityLayer
- Common.Library

Copy Files to Common.Library Project

Right mouse-click on the **Common.Library** project and add a new folder named **HttpClasses**.

Copy all the files **_SampleCode\HttpClasses** folder into the **HttpClasses** folder.

Copy Repository Classes

Right mouse-click on the **AdventureWorks.DataLayer.API** project and add a new folder named **RepositoryClasses**.

Copy all ***Repository.cs** files from the **_SampleCode\RepositoryClasses-API** folder into this new folder.

Lab 2: Change Data Layer Projects in MAUI Project

Open the **AdventureWorks.MAUI\Dependencies\net8.0-windows*\Projects** folder and remove the **AdventureWorks.DataLayer.Mock** reference.

Add a new project reference to the **AdventureWorks.DataLayer.API** project.

Lab 3: Set Minimal Web API Address

Open the **MauiProgram.cs** file and add a new method.

```
#region RegisterHttpClient Method
/// <summary>
/// Create an instance of an HttpClient to use for all
Web API calls
/// </summary>
public static MauiAppBuilder RegisterHttpClient(this
MauiAppBuilder mauiAppBuilder, string baseAddress)
{
    HttpClient client = new()
    {
        BaseAddress = new Uri(baseAddress)
    };

    mauiAppBuilder.Services.AddSingleton<HttpClient>(client)
    ;

    return mauiAppBuilder;
}
#endregion
```

After the code where you added the services, type in the following code.

```
// Add HttpClient with base address for Web API calls
if (DeviceInfo.Platform == DevicePlatform.WinUI) {
    RegisterHttpClient(builder, "http://localhost:0000");
}
else if (DeviceInfo.Platform == DevicePlatform.Android)
{
    RegisterHttpClient(builder, "http://10.0.2.2:0000");
}
```

Open the **AdventureWorks.MinWebAPI\Properties\launchSettings.json** file and get the **port** number in the **http address** located under the **profiles.http.applicationUrl**.

Type in this port number where the **0000**'s are located in the above code you added in the **MauiProgram.cs** file.

Run Multiple Projects

Right mouse-click on the **Solution** and choose **Properties** from the menu.

Under **Common Properties | Startup Project...**

Select the **Multiple Startup Projects** radio button.

Move the **AdventureWorks.MinWebAPI** project to the top of the list.

Set its **Action** to 'Start'.

Move the **AdventureWorksTraining.MAUI** project to the second one in the list.

Set its **Action** to 'Start'.

Click on the **Configuration Properties...** tab.

Check the **Deploy** button next to the **AdventureWorks.MAUI** project, if it is not already checked.

Click the **OK** button.

Try It Out

Run the application on Windows.

Click on the **Users** menu.

Click on one of the Users to display the edit screen.

Ensure all data is showing up correctly.

Lab 4: Run on Android Emulator (Optional)

Right mouse-click on the **\Platforms\Android\Resources** folder and add a new folder named **xml**.

Right mouse-click on this new **xml** folder and add a new XML file named **network_security_config.xml**. Replace the entire contents of this new file with the following XML.

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
  <domain-config cleartextTrafficPermitted="true">
    <domain includeSubdomains="true">10.0.2.2</domain>
  </domain-config>
</network-security-config>
```

Bring up the properties window on this new XML file and set the **Build Action** to **AndroidResource**.

Right mouse-click on the **\Platforms\Android\AndroidManifest.xml** file and select **Open With... | XML (Text) Editor** and click **OK**.

Add the following attribute just within the `<application...>` tag.

```
<application
  android:networkSecurityConfig="@xml/network_security_con
  fig" ... />
```

Try It Out

Run the application on the Android Emulator.

Click on the **Users** menu.

If it does not work, it may be because you do not have a development certificate.

More Information

<https://learn.microsoft.com/en-us/dotnet/maui/data-cloud/local-web-services>