Calling Web API & CORS Lab

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

Lab 1: Create MVC Website

Startup Visual Studio 2022 and select **Create New Project** as shown in Figure 1.

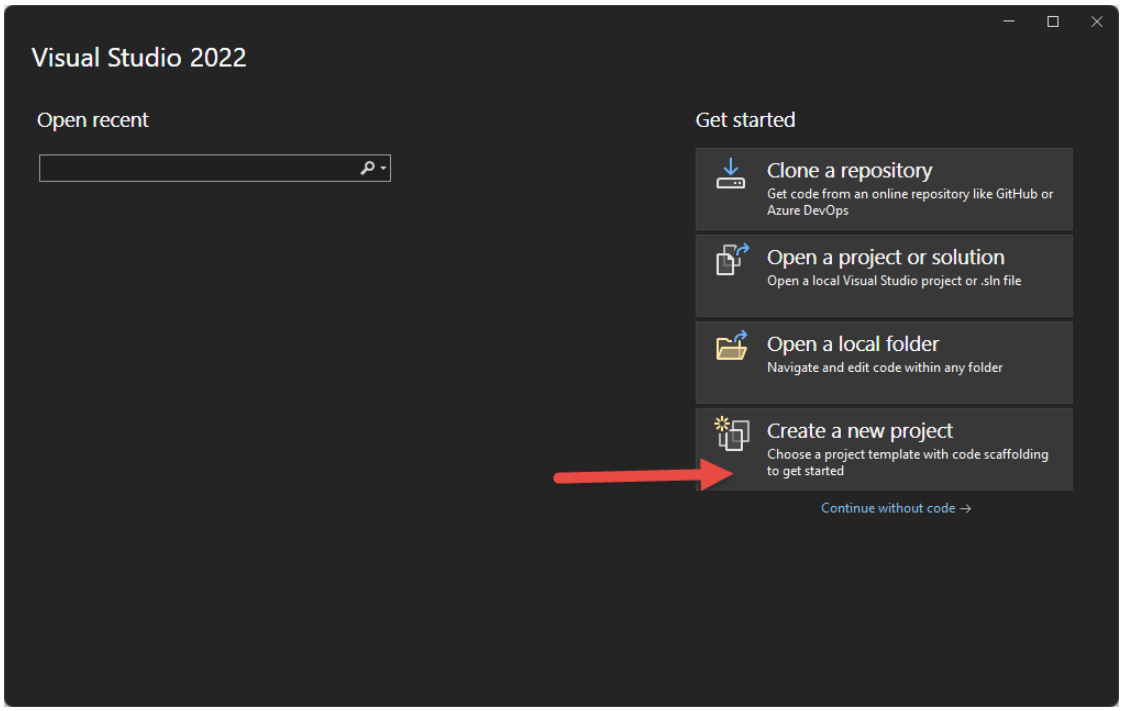

Figure 1: Select what you want to do in Visual Studio startup screen

Create a New Project Screen

Locate the project template **ASP.NET Core Web App (Model-View-Controller)** and select that one as shown in Figure 2.

Click the **Next** button to continue to the next screen

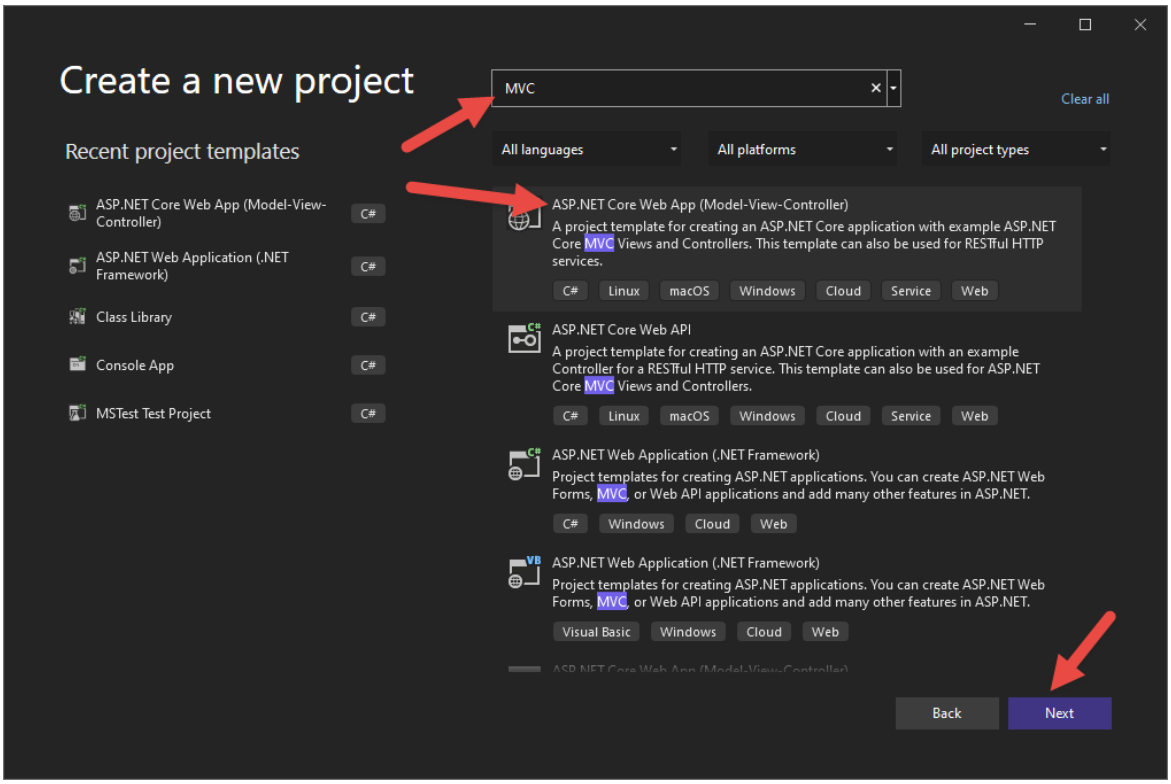

Figure 2: Select the ASP.NET Core Web App (Model-View-Controller) Project.

Configure Your New Project Screen

Set the **Project Name** with **AdvWorks**

Set the **Location** to where you want the project to reside.

Check the Place solution and project in the same directory check box as shown in Figure 3.

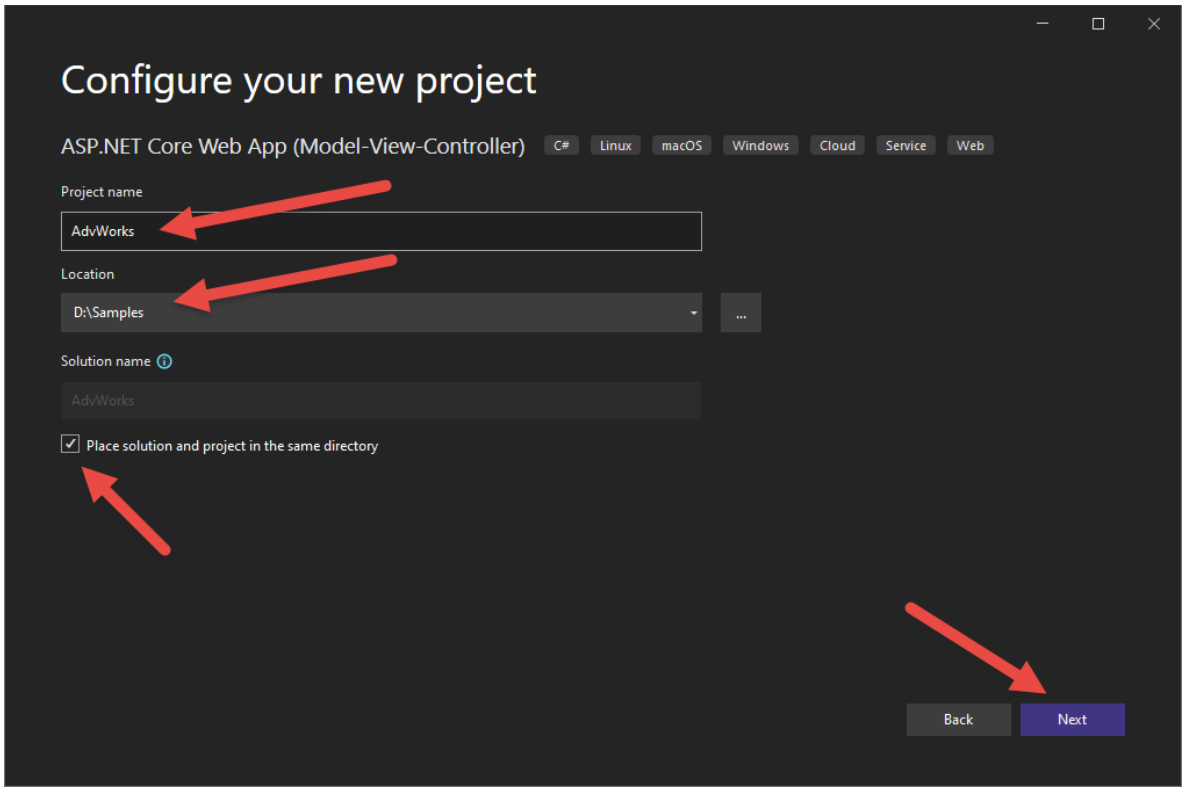

Figure 3: Configure your new project

Additional Information Screen

Choose .NET 6.0 (Long-term support)

Choose Authentication Type = None

Uncheck Configure for HTTPS

Click the **Create** button

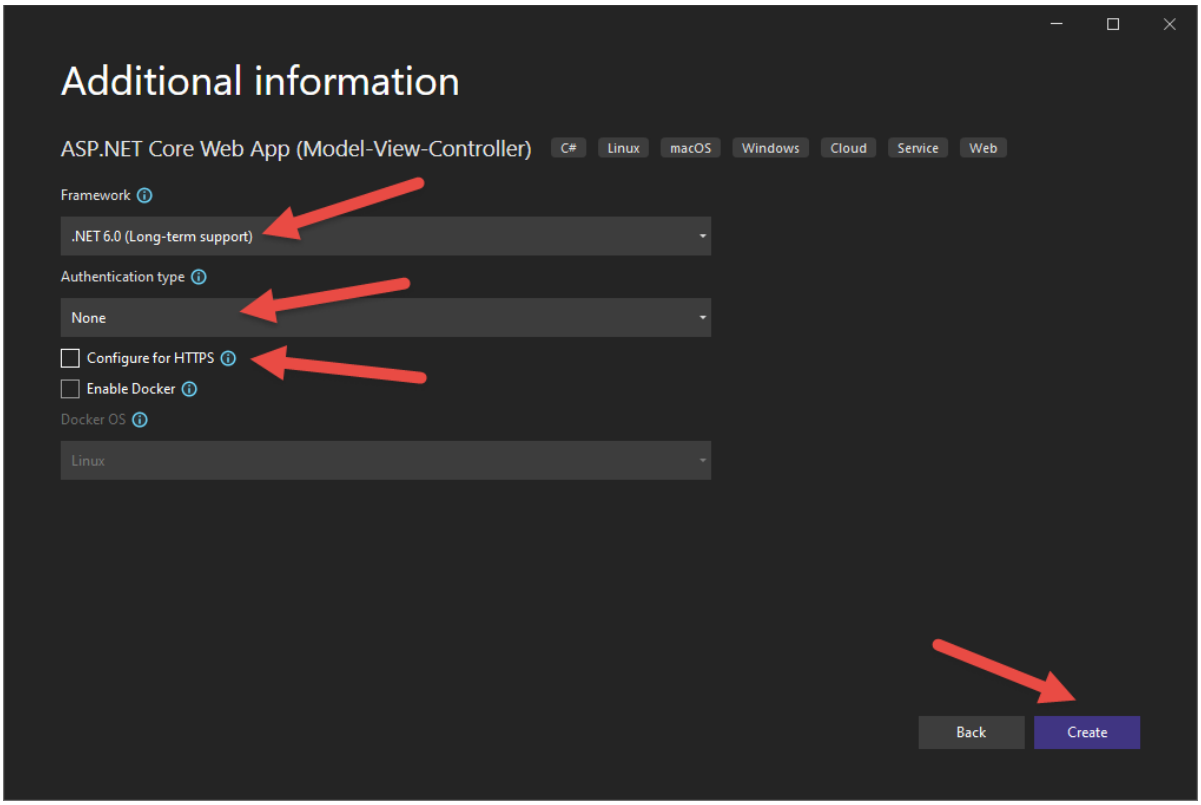

Figure 4: Set the project information

Try it Out

Select Debug | Start Debugging (F5) from the VS menu to build the MVC project and launch a browser.

Lab 2: Call Web API from MVC Razor Page

In the MVC project, open the **Index.cshtml** file and replace the entire contents with the following code.

```
@{
  ViewData["Title"] = "Get a Customer via Web API";
}

<div class="row text-center">
  <div class="col">
    <h1>Get a Customer via Web API</h1>
  </div>
</div>

<div class="row text-center ">
  <div class="col">
    <button class="btn btn-primary"
onclick="getCustomer();">Get a Customer</button>
  </div>
</div>

<div class="row text-center">
  <div class="col">
    <textarea id="customer" rows="10"
cols="100"></textarea>
  </div>
</div>

@section Scripts
{
  <script>
    const URL = "http://localhost:5114/api/Customer/5";

    function getCustomer() {
      fetch(URL)
        .then(response => response.json())
        .then(data => {
          $("#customer").val(JSON.stringify(data));
          console.log(data);
        })
        .catch(error => {
          console.error(error);
          alert("ERROR: Check the Console Window");
        });
    }
  </script>
}
```

NOTE:	Replace the **PORT** number with the value from the Web API Project.

Try it Out

Run the **Web API** application.

Run the **MVC** application.

Click on the **Get a Customer** button on the home page.

You should now see an error alert appear.

Press **F12** to bring up the developer tools in your browser and look at the **Console** window for the CORS error.

Lab 3: Add CORS

Modify Web API Project to Support CORS

Go to the **AdvWorksAPI** project and stop it from running.

Open the **Program.cs** file and after the code where you configured SeriLog add the following code. You need to replace the PORT number with the one from your MVC project.

```
// Add & Configure CORS
builder.Services.AddCors(options =>
{
  options.AddPolicy("AdvWorksAPICorsPolicy",
    builder =>
    {
      builder.WithOrigins("http://localhost:5003");
    });
});
```

NOTE:	Replace the **PORT** number with the value from the MVC Project.

Add the following code **before** the app.Services.CreateScope() call.

```
// Enable CORS Middleware
app.UseCors("AdvWorksAPICorsPolicy");
```

Try it Out

Run the **Web API** application.

Run the **MVC** application.

Click on the **Get a Customer** button on the home page.

You should now see the customer data appear.