

# .NET 9 App Dev Hands-On Lab

## Blazor Lab 1 – The Blazor Projects

This lab walks you through creating the Blazor Web Assembly (WASM) projects and adding/updating the NuGet packages.

### Part 1: Creating the Projects

#### Step 1: Create the Projects

Note: Non-windows users must adjust the directory separator using the following commands.

Note: PowerShell and bash need quotes around the version monikers.

- Create the Class Library for the entities and add it to the solution:  
**NOTE:** Using PowerShell, the version intervals must be surrounded by single quotes (like '[17.\*,18.0)'). Run the commands as shown here if using a regular command prompt.

[Windows]

```
dotnet new classlib -lang c# -n AutoLot.Blazor.Models -o .\AutoLot.Blazor.Models -f net9.0
dotnet sln AutoLot.sln add AutoLot.Blazor.Models
dotnet add AutoLot.Blazor.Models package Microsoft.VisualStudio.Threading.Analyzers -v [17.*,18.0)
```

- Create the Blazor WebAssembly Standalone App project, add it to the solution, and add a reference to the class library:

```
dotnet new blazorwasm -lang c# -au none -n AutoLot.Blazor -o .\AutoLot.Blazor -f net9.0
dotnet sln AutoLot.sln add AutoLot.Blazor
dotnet add AutoLot.Blazor reference AutoLot.Blazor.Models
```

- Add the required NuGet packages to the AutoLot.Blazor project (each on only one line):

```
dotnet add AutoLot.Blazor package Microsoft.AspNetCore.Components.WebAssembly -v [9.0.*,10.0)
dotnet add AutoLot.Blazor package Microsoft.AspNetCore.Components.WebAssembly.DevServer -v [9.0.*,10.0)
dotnet add AutoLot.Blazor package Microsoft.Extensions.Http -v [9.0.*,10.0)
dotnet add AutoLot.Blazor package Microsoft.Extensions.Options.ConfigurationExtensions -v [9.0.*,10.0)
dotnet add AutoLot.Blazor package Microsoft.VisualStudio.Threading.Analyzers -v [17.*,18.0)
dotnet add AutoLot.Blazor package Microsoft.Web.LibraryManager.Build -v [2.*,3.0)
```

#### Step 2: Disable Nullable Reference Types

- Open the new project files (AutoLot.Blazor.csproj, AutoLot.Blazor.Models.csproj) and update the PropertyGroup to disable nullable reference types. NOTE: The order is different in the two project files. Be careful to disable Nullable and NOT ImplicitUsings:

<Nullable>**disable**</Nullable>

- Open the NavMenu.Razor file and update the string property not to be nullable:

```
private string NavMenuCssClass => collapseNavMenu ? "collapse" : null;
```

### Step 3: (VS) Set AutoLot.Blazor as the startup project

Right-click on AutoLot.Blazor and select "Set as Startup Project" from the context menu.

### Step 4: Adjust the launchsettings.json file

Open the launchsettings.json file (in the Properties directory of the project) and move the HTTPS profile to the top. **NOTE:** If you want to add the IIS Profile, select IIS Express in Visual Studio, and one will be created.

## Part 3: Clean up Unnecessary Scaffolded Code

- Delete Pages\Counter.razor and Pages\Weather.razor files.
- Delete the wwwroot\sample-data folder and the JSON file it contains.
- Delete the following from Layout\NavMenu.razor:

```
<div class="nav-item px-3">
  <NavLink class="nav-link" href="counter">
    <span class="bi bi-plus-square-fill-nav-menu" aria-hidden="true"></span> Counter
  </NavLink>
</div>
<div class="nav-item px-3">
  <NavLink class="nav-link" href="weather">
    <span class="bi bi-list-nested-nav-menu" aria-hidden="true"></span> Weather
  </NavLink>
</div>
```

## Part 4: Manage Client-Side Libraries

- Add a JSON file named `libman.json` to the root of the **AutoLot.Blazor** project. Update the file to match the following:

```
{
  "version": "1.0",
  "defaultProvider": "cdnjs",
  "libraries": [
    {
      "library": "twitter-bootstrap@5.3.3",
      "destination": "wwwroot/lib/bootstrap",
      "files": [
        "css/bootstrap.css",
        "css/bootstrap.min.css"
      ]
    },
    {
      "library": "font-awesome@6.7.1",
      "destination": "wwwroot/lib/font-awesome/",
      "files": [
        "css/all.min.css",
        "css/all.css",
        "sprites/regular.svg",
        "sprites/solid.svg",
        "webfonts/fa-solid-900.ttf",
        "webfonts/fa-solid-900.woff2",
        "webfonts/fa-regular-400.ttf",
        "webfonts/fa-regular-400.woff2"
      ]
    }
  ]
}
```

- Delete the `wwwroot\lib\bootstrap\dist` folder from **AutoLot.Blazor**. Right-click on the `libman.json` file and select “Restore Client-Side Libraries”.
- Update the `wwwroot\Index.html` by replacing this line:

```
<link rel="stylesheet" href="lib/bootstrap/dist/bootstrap.min.css" />
```

- With these lines:

```
<link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet" />
<link href="lib/font-awesome/css/all.min.css" rel="stylesheet" />
```

## Summary

This lab created the Blazor Wasm project and the shared class library for the models.

## Next steps

In the next part of this tutorial series, you will start building the Blazor application.