

# .NET 10 App Dev Hands-On Lab

## Blazor Lab 1 – The Blazor Projects

This lab guides you through creating Blazor Web Assembly (WASM) projects and adding or updating NuGet packages.

### Part 1: Creating the Projects

#### Step 1: Create the Projects

Note: Non-Windows users must adjust the directory separator when using the following CLI 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:** When using PowerShell, the version intervals must be surrounded by single quotes (e.g., '[17.\*,19.0)'). 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 net10.0
dotnet sln AutoLot.sln add AutoLot.Blazor.Models
dotnet add AutoLot.Blazor.Models package Microsoft.VisualStudio.Threading.Analyzers -v [17.*,19.0)
dotnet add AutoLot.Blazor.Models package Microsoft.Extensions.Validation -v '[10.*,11.0)'
```

- 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 net10.0
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 '[10.*,11.0)'
dotnet add AutoLot.Blazor package Microsoft.AspNetCore.Components.WebAssembly.DevServer -v '[10.*,11.0)'
dotnet add AutoLot.Blazor package Microsoft.Extensions.Validation -v '[10.*,11.0)'
dotnet add AutoLot.Blazor package Microsoft.Extensions.Http -v '[10.*,11.0)'
dotnet add AutoLot.Blazor package Microsoft.Extensions.Options.ConfigurationExtensions -v '[10.*,11.0)'
dotnet add AutoLot.Blazor package Microsoft.VisualStudio.Threading.Analyzers -v '[17.*,18.0)'
dotnet add AutoLot.Blazor package Microsoft.Web.LibraryManager.Build -v '[3.*,4.0)'
```

## Step 2: Disable Nullable Reference Types and Set Environment Value

### Coplot Agent Mode

Prompt: Update the AutoLot.Blazor and AutoLot.Blazor.Models projects to set nullable reference types to disable. Update the NavMenu.CssClass property in NavMenu.Razor to not be nullable. Add the following property groups to the AutoLot.Blazor.csproj:

```
<PropertyGroup Condition="'$(Configuration)' == 'Debug'">
  <WasmApplicationEnvironmentName>Development</WasmApplicationEnvironmentName>
</PropertyGroup>
<PropertyGroup Condition="'$(Configuration)' == 'Release'">
  <WasmApplicationEnvironmentName>Production</WasmApplicationEnvironmentName>
</PropertyGroup>
```

### Manual

- Open the AutoLot.Blazor.csproj file and update the PropertyGroup to disable nullable reference types:

```
<Nullable>disable</Nullable>
```

- Add the environment property groups:

```
<PropertyGroup Condition="'$(Configuration)' == 'Debug'">
  <WasmApplicationEnvironmentName>Development</WasmApplicationEnvironmentName>
</PropertyGroup>
<PropertyGroup Condition="'$(Configuration)' == 'Release'">
  <WasmApplicationEnvironmentName>Production</WasmApplicationEnvironmentName>
</PropertyGroup>
```

- Open the AutoLot.Blazor.Models.csproj file and update the PropertyGroup to disable nullable reference types.

```
<Nullable>disable</Nullable>
```

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

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

## 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. Delete the HTTP profile. **NOTE:** If you want to add the IIS Profile, select IIS Express in Visual Studio, run the app, and one will be created.

## Part 2: Clean up Unnecessary Scaffolded Code

### Coplot Agent Mode

Prompt: In the AutoLot.Blazor project, delete Pages\Counter.razor and Pages\Weather.razor. Delete the wwwroot\sample-data folder and its contents. Update Layout\NavMenu.razor by removing the following code:

```
<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>
```

### Manual

- 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 3: 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": "3.0",
  "defaultProvider": "cdnjs",
  "libraries": [
    {
      "library": "twitter-bootstrap@5.3.8",
      "destination": "wwwroot/lib/bootstrap",
      "files": [
        "css/bootstrap.css",
        "css/bootstrap.min.css"
      ]
    },
    {
      "library": "font-awesome@6.7.2",
      "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 rel="stylesheet" href="lib/bootstrap/css/bootstrap.min.css" />
<link rel="stylesheet" href="lib/font-awesome/css/all.min.css" />
```

- Update the `NavMenu.razor` component to use the Font Awesome font for the Home link (changes in bold):

```
<div class="nav-item px-3">
  <NavLink class="nav-link" href="" Match="NavLinkMatch.All">
    <span class="fa-solid fa-home pe-2" aria-hidden="true"></span>Home
  </NavLink>
</div>
```

## 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.