.NET 9 App Dev Hands-On Lab

API Lab 1 – Solution and Projects

This lab walks you through creating the ASP.NET Core RESTful service project and adding/updating the NuGet packages. Before starting this lab, you must have completed EF Lab 01.

Part 1: Creating the Project

The API labs build onto the current `AutoLot` solution.

Step 1: Create the ASP.NET Core RESTful Service Project

- Create the ASP.NET Core service application project:

```
[Windows]
dotnet new webapi -lang c# -n AutoLot.Api -au none -o .\AutoLot.Api --use-controllers -f net9.0
```

- Add the project to the solution and project references:

```
dotnet sln AutoLot.sln add AutoLot.Api
dotnet add AutoLot.Api reference AutoLot.Models
dotnet add AutoLot.Api reference AutoLot.Dal
dotnet add AutoLot.Api reference AutoLot.Services
```

- Add the required NuGet packages to the project:

```
dotnet add AutoLot.Api package Asp.Versioning.Mvc -v '[8.*,10.0)'
dotnet add AutoLot.Api package Asp.Versioning.Mvc.ApiExplorer -v '[8.*,10.0)'
dotnet add AutoLot.Api package Microsoft.EntityFrameworkCore.SqlServer -v '[9.0.*,10.0)'
dotnet add AutoLot.Api package Microsoft.EntityFrameworkCore.Design -v '[9.0.*,10.0)'
dotnet add AutoLot.Api package Microsoft.VisualStudio.Threading.Analyzers -v '[17.*,18.0)'
dotnet add AutoLot.Api package Microsoft.VisualStudio.Web.CodeGeneration.Design -v '[9.0.*,10.0)'
dotnet add AutoLot.Api package Swashbuckle.AspNetCore -v '[6.*,7.0)'
dotnet add AutoLot.Api package Swashbuckle.AspNetCore.Annotations -v '[6.*,7.0)'
dotnet add AutoLot.Api package Swashbuckle.AspNetCore.Swagger -v '[6.*,7.0)'
dotnet add AutoLot.Api package Swashbuckle.AspNetCore.SwaggerGen -v '[6.*,7.0)'
dotnet add AutoLot.Api package Swashbuckle.AspNetCore.SwaggerUI -v '[6.*,7.0)'
dotnet add AutoLot.Api package System.Text.Json -v '[9.0.*,10.0)'
```

Part 2: Disable Nullable Reference Types and Check Globalization

- Open the project file (*.csproj) for the new projects and disable nullable types:

```
<PropertyGroup>
  <TargetFramework>net9.0</TargetFramework>
  <Nullable>disable</Nullable>
  <ImplicitUsings>enable</ImplicitUsings>
</PropertyGroup>
```

- Older versions of the API template added a node for Invariant Globalization Types. If this node exists in your project, delete it or set the value to false:

```
<InvariantGlobalization>false</InvariantGlobalization>
```

Part 3: Adjust the launchsettings.json file

- Move the https profile to the top in `AutoLot.Api`

If a profile isn't selected, the first profile in the list will be selected by default. Move the `https` profile to the top so it gets selected, not the `http` profile.
NOTE: The IIS profile will not show unless you run the project using IIS in Visual Studio.

- Update the ports in the `launchSettings.json` file to 5011 for https and 5010 for http (only relevant parts shown here):

```
"iisExpress": {
  "applicationUrl": "http://localhost:5010",
  "launchBrowser":true,
  "launchUrl": "swagger",
  "sslPort": 5011
}
"https": {
  //omited for brevity
  "applicationUrl": "https://localhost:5011;http://localhost:5010",
  "launchBrowser":true,
  "launchUrl": "swagger",
},
```

Part 4: Remove Extra Scaffolded Files

- Remove the following scaffolded files:

```
Controllers/WeatherForecaseController.cs
WeatherForecast.cs
```

Summary

This lab created the solution and the API project, added the NuGet packages and the appropriate references, and finally updated the project settings.

Next steps

In the next part of this tutorial series, you will start working with the Web API project.