

.NET 10 App Dev Hands-On Lab

API Lab 1 – Solution and Projects

This lab guides you through creating an ASP.NET Core RESTful service project and updating the NuGet packages. Before starting this lab, you must have completed EF Core Lab 06.

Part 1: Creating the Project

The API labs build on the current AutoLot solution. Run all of the commands in the solution folder.

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 net10.0
```

- 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.AspNetCore.OpenApi -v [10.*,11.0)
dotnet add AutoLot.Api package Microsoft.EntityFrameworkCore.SqlServer -v [10.*,11.0)
dotnet add AutoLot.Api package Microsoft.EntityFrameworkCore.Design -v [10.*,11.0)
dotnet add AutoLot.Api package Microsoft.VisualStudio.Threading.Analyzers -v [17.*,19.0)
dotnet add AutoLot.Api package Swashbuckle.AspNetCore.SwaggerUI -v [10.*,11.0)
dotnet add AutoLot.Api package Scalar.AspNetCore -v [2.*,3)
```

```
dotnet add AutoLot.Mvc package Microsoft.Build
```

Part 2: Disable Nullable Reference

Coplot Agent Mode

Prompt: Update the AutoLot.Api project to set nullable reference types to disable.

Manual

- Open the new project file (*.csproj) update the Nullable node of the PropertyGroup to the following (change is in bold):

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

Part 3: Adjust the launchsettings.json file

- Move the https profile to the top in AutoLot.Api and delete the http profile.

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 and add the launch URL for the OpenAPI document (only relevant parts shown here):

```
"profiles": {
  "https": {
    //omited for brevity
    "applicationUrl": "https://localhost:5011;http://localhost:5010",
    "launchBrowser": true,
    "launchUrl": "openapi/v1.json",
  },
  "IIS Express": {
    "commandName": "IISExpress",
    "launchBrowser": true,
    "launchUrl": "openapi/v1.json",
    "environmentVariables": {
      "ASPNETCORE_ENVIRONMENT": "Development"
    }
  }
},
"iisSettings": {
  "windowsAuthentication": false,
  "anonymousAuthentication": true,
  "iisExpress": {
    "applicationUrl": "http://localhost:5010/",
    "sslPort": 5011
  }
}
```

Part 4: Remove Extra Scaffolded Files

- Remove the following scaffolded files:

```
Controllers/WeatherForecastController.cs
WeatherForecast.cs
```

Summary

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

Next steps

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