.NET 10 App Dev Hands-On Lab

Razor Pages with API Lab 1b – API Data Services

This lab adds the HTTP Client factory for web applications to leverage the RESTful service. Before starting this lab, you must have completed Razor Pages/MVC with API Lab 1a.

Copilot Agent Mode

```
Setup Prompt: Always use file scoped namespaces. Always combine attributes on a single line when
possible. The project does not use nullable reference types. There is a GlobalUsings.cs file that
includes common usings, don't include using statements in new files if they are already in the
globalusings.cs file. I prefer expression bodied members when possible. Single line if statements
should still use braces. Use ternary operators when appropriate.  Use internal over private. All
classes and methods are public unless told otherwise. Use primary constructors when possible and
don't declare a class level variable if the parameter from the constructor can be used.
All work is to be done in the AutoLot.Web project.

Prompt: Add the following global usings to the GlobalUsings.cs file if they do not already exist
(sorted alphabetically. Don't remove any existing global using statements).
global using AutoLot.Services.ApiWrapper.Configuration;
global using AutoLot.Services.DataServices.Api;

Prompt: In Program.cs, comment out the existing service registrations for ICarDataService and
IMakeDataService
builder.Services.AddScoped<ICarDataService, CarDalDataService>();
builder.Services.AddScoped<IMakeDataService, MakeDalDataService>();

Prompt: In Program.cs, add the following:
if (builder.Configuration.GetValue<bool>("UseApi"))
{
  builder.Services.AddScoped<ICarDataService, CarApiDataService>();
  builder.Services.AddScoped<IMakeDataService, MakeApiDataService>();
}
else
{
  builder.Services.AddScoped<ICarDataService, CarDalDataService>();
  builder.Services.AddScoped<IMakeDataService, MakeDalDataService>();
}

Prompt: Add the call to ConfigureApiServiceWrapper in Program.cs as follows:
builder.Services.ConfigureApiServiceWrapper(builder.Configuration);

Prompt: In appsettings.Development.json, change the rebuildDataBase node to false.

Prompt: In appsettings.Development.json, add a node as follows:
"UseApi": true,
Prompt: In appsettings.Development.json, add a new section named ApiServiceSettings with the
following values:
"ApiServiceSettings": {
    "Uri": "https://localhost:5011/",
    "CarBaseUri": "api/cars",
    "MakeBaseUri": "api/makes",
    "MajorVersion": 1,
    "MinorVersion": 0,
    "Status": ""
  }

Prompt: In appsettings.Staging.json, change the rebuildDataBase node to false.

Prompt: In appsettings.Staging.json, add a node as follows:
"UseApi": true,

Prompt: In appsettings.Staging.json, add a new section named ApiServiceSettings with the following
values:
"ApiServiceSettings": {
    "Uri": "https://localhost:5011/",
    "CarBaseUri": "api/cars",
    "MakeBaseUri": "api/makes",
    "MajorVersion": 1,
    "MinorVersion": 0,
    "Status": ""
  }
```

Manual

Step 1: Update the GlobalUsings.cs file

- In the AutoLot.Web application, add the following to the GlobalUsings.cs file:

```
global using AutoLot.Services.ApiWrapper.Configuration;
global using AutoLot.Services.DataServices.Api;
```

Step 2: Register the Api Data Services

- Update the call to add the data services in the Program.cs class to the following:

~~builder.Services.AddScoped<ICarDataService, CarDalDataService>();~~
~~builder.Services.AddScoped<IMakeDataService, MakeDalDataService>();~~

```
if (builder.Configuration.GetValue<bool>("UseApi"))
{
  builder.Services.AddScoped<ICarDataService, CarApiDataService>();
  builder.Services.AddScoped<IMakeDataService, MakeApiDataService>();
}
else
{
  builder.Services.AddScoped<ICarDataService, CarDalDataService>();
  builder.Services.AddScoped<IMakeDataService, MakeDalDataService>();
}
```

- Add the call to `ConfigureApiServiceWrapper` in the top-level statements in the `Program.cs`:

```
builder.Services.ConfigureApiServiceWrapper(builder.Configuration);
```

Step 3: Update the Application Settings

- Update the `RebuildDatabase` flag to false:

```
"RebuildDataBase": false,
```

- Add the following to the `appsettings.Development.json` and `appsettings.Staging.json` files (don't forget to add a comma after the last line before adding the new lines):

```
"UseApi": true,
"ApiServiceSettings": {
  "Uri": "https://localhost:5011/",
  "CarBaseUri": "api/Cars",
  "MakeBaseUri": "api/Makes",
  "MajorVersion": 1,
  "MinorVersion": 0,
  "Status": ""
}
```

Step 4: Test the Apps

- Using Visual Studio, right-click the solution and set the `AutoLot.Api` and `AutoLot.Web` projects to be the startup projects.
- Using Visual Studio Code, use dotnet run for both apps, starting with `AutoLot.Api`.

Summary

This lab changed the data source to the RESTful services.