.NET App Dev Hands-On Lab

Razor Pages with API Lab 1b – API Data Services

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

Part 1: Use the API Data Service Classes

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 1: Register the Api Data Services

- Update the call to add the data services to 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>();
}
```

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

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

Step 2: Update the Application Settings

- Add the following to the `appsettings.Development.json` file. When set to true, the API will be used for CRUD operations, when set to false, the repos will be used:

```
  "UseApi": true,
  "ApiServiceSettings": {
    "Uri": "https://localhost:5011/",
    "CarBaseUri": "api/v1/Cars",
    "MakeBaseUri": "api/v1/Makes",
    "MajorVersion": 1,
    "MinorVersion": 0,
    "Status": ""
  },
```

Step 3: 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 watch run for both apps, starting with `AutoLot.Api`.

Summary

This lab added `ApiDataServices` to `AutoLot.Web`.