

# .NET App Dev Hands-On Workshop

## Blazor Lab 4 – Data Services

This lab adds data services to the AutoLot.Blazor project. Before starting this lab, you must have completed Blazor Lab 3.

### Part 1: Add the Data Service and Interfaces

#### Step 1: Add the service interfaces.

- Add the following to the GlobalUsings.cs file in the AutoLot.Blazor project:

```
global using AutoLot.Blazor.Models.Entities;  
global using AutoLot.Blazor.Models.Entities.Base;  
global using AutoLot.Blazor.Models.ViewModels;
```

- In the Services folder in the AutoLot.Blazor project, add a new folder named Interfaces. In this folder, add a new folder named Base and add a new interface named IDataServiceBase and update it to the following:

```
namespace AutoLot.Blazor.Services.Interfaces;  
public interface IDataServiceBase<TEntity> where TEntity : BaseEntity  
{  
    Task<TEntity> GetEntityAsync(int id);  
    Task<TEntity> AddEntityAsync(TEntity entity);  
    Task<TEntity> UpdateEntityAsync(int id, TEntity entity);  
    Task DeleteEntityAsync(TEntity entity);  
    Task<List<TEntity>> GetAllEntitiesAsync();  
}
```

- Add the following to the GlobalUsings.cs file:

```
global using AutoLot.Blazor.Services.Interfaces;  
global using AutoLot.Blazor.Services.Interfaces.Base;
```

- In the Interfaces folder, add two interface files: ICarDataService.cs and IMakeDataService.cs. Update the code to the following listings:

```
//ICarDataService.cs  
namespace AutoLot.Blazor.Services.Interfaces;  
public interface ICarDataService : IDataServiceBase<Car>  
{  
    Task<List<Car>> GetByMakeAsync(int makeId);  
}
```

```
//IMakeDataService.cs  
namespace AutoLot.Blazor.Services.Interfaces;  
public interface IMakeDataService : IDataServiceBase<Make>  
{  
}
```

## Step 2: Add the BaseDataService class

- Create a folder named Base in the Services folder. In the Base folder, add a new class named BaseDataService.cs. Update the code to the following:

```
namespace AutoLot.Blazor.Services.Base;
public class BaseDataService
{
    protected static List<Make> Makes =
    [
        new() { Id = 1, Name = "VW" },
        new() { Id = 2, Name = "Ford" },
        new() { Id = 3, Name = "Saab" },
        new() { Id = 4, Name = "Yugo" },
        new() { Id = 5, Name = "BMW" },
        new() { Id = 6, Name = "Pinto" }
    ];
    protected List<Car> CarList =
    [
        new() { Id = 1, MakeId = 1, Color = "Black", PetName = "Zippy", Price = "$45,000.00",
            MakeNavigation = Makes.First(m => m.Id == 1) },
        new() { Id = 2, MakeId = 2, Color = "Rust", PetName = "Rusty", Price = "$45,000.00",
            MakeNavigation = Makes.First(m => m.Id == 2) },
        new() { Id = 3, MakeId = 3, Color = "Black", PetName = "Mel", Price = "$45,000.00",
            MakeNavigation = Makes.First(m => m.Id == 3) },
        new() { Id = 4, MakeId = 4, Color = "Yellow", PetName = "Clunker", Price = "$45,000.00",
            MakeNavigation = Makes.First(m => m.Id == 4) },
        new() { Id = 5, MakeId = 5, Color = "Black", PetName = "Bimmer", Price = "$45,000.00",
            MakeNavigation = Makes.First(m => m.Id == 5) },
        new() { Id = 6, MakeId = 5, Color = "Green", PetName = "Hank", Price = "$45,000.00",
            MakeNavigation = Makes.First(m => m.Id == 5) },
        new() { Id = 7, MakeId = 5, Color = "Pink", PetName = "Pinky", Price = "$45,000.00",
            MakeNavigation = Makes.First(m => m.Id == 5) },
        new() { Id = 8, MakeId = 6, Color = "Black", PetName = "Pete", Price = "$45,000.00",
            MakeNavigation = Makes.First(m => m.Id == 6) },
        new() { Id = 9, MakeId = 4, Color = "Brown", PetName = "Brownie", Price = "$45,000.00",
            MakeNavigation = Makes.First(m => m.Id == 4) },
        new() { Id = 10, MakeId = 1, Color = "Rust", PetName = "Lemon", IsDrivable = false,
            Price = "$45,000.00", MakeNavigation = Makes.First(m => m.Id == 1) }
    ];
}
```

- Add the following to the GlobalUsings.cs file:

```
global using AutoLot.Blazor.Services.Base;
```

### Step 3: Add the Car data service implementation.

- Create a new class named CarDataService in the Services folder and update it to the following:

```
namespace AutoLot.Blazor.Services;
public class CarDataService : BaseDataService, ICarDataService
{
    public async Task<Car> GetEntityAsync(int id)
        => await Task.FromResult(CarList.FirstOrDefault(c => c.Id == id));

    public async Task<Car> AddEntityAsync(Car entity)
    {
        entity.Id = CarList.Max(x => x.Id) + 1;
        entity.MakeNavigation = Makes.First(m => m.Id == entity.MakeId);
        CarList.Add(entity);
        return await Task.FromResult(entity);
    }

    public async Task<Car> UpdateEntityAsync(int id, Car entity)
    {
        entity.MakeNavigation = Makes.First(m => m.Id == entity.MakeId);
        return await Task.FromResult(entity);
    }

    public async Task DeleteEntityAsync(Car entity)
    {
        var carToRemove = CarList.FirstOrDefault(c => c.Id == entity.Id);
        if (carToRemove is not null)
        {
            CarList.Remove(carToRemove);
        }
        await Task.CompletedTask;
    }

    public async Task<List<Car>> GetAllEntitiesAsync() => await Task.FromResult(CarList);

    public async Task<List<Car>> GetByMakeAsync(int makeId)
        => await Task.FromResult(CarList.Where(x => x.MakeId == makeId).ToList());
}
```

## Step 4: Add the Make data service implementation.

- Add a new class named `MakeDataService.cs` in the `Services` folder and update the code to the following:

```
namespace AutoLot.Blazor.Services;
public class MakeDataService : BaseDataService, IMakeDataService
{
    public async Task<Make> GetEntityAsync(int id)
        => await Task.FromResult(Makes.FirstOrDefault(c => c.Id == id));
    public async Task<Make> AddEntityAsync(Make entity)
    {
        entity.Id = Makes.Max(x => x.Id)+1;
        Makes.Add(entity);
        return await Task.FromResult(entity);
    }
    public async Task<Make> UpdateEntityAsync(int id, Make entity) => await Task.FromResult(entity);
    public async Task DeleteEntityAsync(Make entity)
    {
        var carToRemove = Makes.FirstOrDefault(c => c.Id == entity.Id);
        if (carToRemove is not null)
        {
            Makes.Remove(carToRemove);
        }
        await Task.CompletedTask;
    }
    public async Task<List<Make>> GetAllEntitiesAsync() => await Task.FromResult(Makes);
}
```

## Part 2: Configure AutoLot.Blazor

### Step 1: Add the AppSettings files

- Add three json files named `appsettings.json`, `appsettings.Development.json`, and `appsettings.Staging.json` to the wwwroot folder of the AutoLot.Blazor project. Update the files to the following:

```
//appsettings.json
{
  "DealerInfo": {
    "DealerName": "Skimedic's Used Cars",
    "City": "West Chester",
    "State": "Ohio"
  }
}

//appsettings.Development.json
{
  "DealerInfo": {
    "DealerName": "Skimedic's Used Cars Development Site",
    "City": "West Chester",
    "State": "Ohio"
  }
}

//appsettings.Staging.json
{
  "DealerInfo": {
    "DealerName": "Skimedic's Used Cars Staging Site",
    "City": "West Chester",
    "State": "Ohio"
  }
}
```

- Add the following to the `AutoLot.Blazor.csproj` file:

```
<ItemGroup>
  <Content Update="wwwroot\appsettings.json">
    <CopyToOutputDirectory>Always</CopyToOutputDirectory>
  </Content>
  <Content Update="wwwroot\appsettings.*.json">
    <CopyToOutputDirectory>Always</CopyToOutputDirectory>
  </Content>
</ItemGroup>
```

## Step 2: Update the Program.cs

- Add the following three lines to the Program.cs file just before the call to RunAsync():

```
builder.Services.AddScoped<ICarDataService, CarDataService>();  
builder.Services.AddScoped<IMakeDataService, MakeDataService>();  
builder.Services.Configure<DealerInfo>(builder.Configuration.GetSection(nameof(DealerInfo)));  
  
await builder.Build().RunAsync();
```

## Summary

This lab added shared services and configured them into the application using the configuration system.

## Next Steps

The next lab will work with the UI, Pages, and components.