.NET 9 App Dev Hands-On Workshop

Razor Lab 6 – Helper Components

This lab adds helper components to the `AutoLot.Blazor` project. Before starting this lab, you must have completed Blazor Lab 5.

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. If there is a GlobalUsings.cs file,
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. Don't add a constructor unless instructed to do so. Use primary
constructors when possible and don't declare a class level variable if the parameter from the
constructor can be used. Don't initialize properties unless instructed to do so. The @code block
in razor components should be at the bottom of the file. All work is to be done in the
AutoLot.Blazor project.

Prompt: Add a new directory named Helpers to the AutoLot.Blazor project. In this folder, add
another new folder named ItemHelpers. In this folder, add a new Razor component named
CreateHelper.razor. Clear out the contents and update it to the following:
<a href="/@RouteStart/create" class="text-success">
  Create <i class="fa-solid fa-plus"></i></a>
@code {
  [Parameter] public string RouteStart { get; set; }
}

Prompt: Add a new Razor component named DeleteHelper.razor to the ItemHelpers folder. Clear out
the contents and update it to the following:
<a href="/@RouteStart/delete/@Id" class="text-danger">
  Delete <i class="fa-solid fa-trash"></i></a> |
@code {
  [Parameter] public int Id { get; set; }
  [Parameter] public string RouteStart { get; set; }
}

Prompt: Add a new Razor component named DetailsHelper.razor to the ItemHelpers folder. Clear out
the contents and update it to the following:
<a href="/@RouteStart/details/@Id" class="text-info">
   Details <i class="fa-solid fa-info-circle"></i></a> |
@code {
  [Parameter] public int Id { get; set; }
  [Parameter] public string RouteStart { get; set; }
}
```

Prompt: Add a new Razor component named EditHelper.razor to the ItemHelpers folder. Clear out the contents and update it to the following:

```
<a href="/@RouteStart/edit/@Id" class="text-warning">
  Edit <i class="fa-solid fa-edit"></i></a> |
@code {
  [Parameter] public int Id { get; set; }
  [Parameter] public string RouteStart { get; set; }
}
```

Prompt: Add a new Razor component named ListHelper.razor to the ItemHelpers folder. Clear out the contents and update it to the following:

```
<a href="/@RouteStart/index" class="text-primary">
  Back to List <i class="fa-solid fa-list"></i></a>
@code {
  [Parameter] public string RouteStart { get; set; }
}
```

Prompt: Add the following to the _Imports.razor file (don't remove any existing entries):

```
@using AutoLot.Blazor.Helpers
@using AutoLot.Blazor.Helpers.ItemHelpers
```

Prompt: Add a new folder named CarHelpers under the Helpers folder. Add a new Razor component named CarDetail.razor to the CarHelpers folder. Clear out the contents and update it to the following:

```
<h1>Details for @CarInstance.PetName</h1>
<dl class="row">
  <dt class="col-sm-2">Make</dt>
  <dd class="col-sm-10">@CarInstance?.MakeNavigation?.Name</dd>
  <dt class="col-sm-2">Color</dt>
  <dd class="col-sm-10">@CarInstance?.Color</dd>
  <dt class="col-sm-2">Pet Name</dt>
  <dd class="col-sm-10">@CarInstance?.PetName</dd>
  <dt class="col-sm-2">Price</dt>
  <dd class="col-sm-10">@CarInstance?.Price</dd>
  <dt class="col-sm-2">Date Built</dt>
  <dd class="col-sm-10">@CarInstance?.DateBuilt?.ToShortDateString()</dd>
  <dt class="col-sm-2">Is Drivable</dt>
  <dd class="col-sm-10">@CarInstance?.IsDrivable</dd>
</dl>
@code {
  [Parameter]
  [EditorRequired]
  public Car CarInstance { get; set; } = default!;
}
```

Prompt: Add a new Razor component named CarEdit.razor to the CarHelpers folder. Clear out the contents and update it to the following:

```
<DataAnnotationsValidator/>
<ValidationSummary/>
<div>
  <label class="col-form-label" for="@nameof(Car.MakeId)">Make</label>
  <InputSelect class="form-control" @bind-Value="CarInstance.MakeId" id="@nameof(Car.MakeId)">
    @foreach (var item in Makes)
    {
      <option value="@item.Id">@item.Name</option>
    }
  </InputSelect>
  <ValidationMessage For="() => CarInstance.MakeId"/>
</div>
<div>
  <label class="col-form-label" for="@nameof(Car.Color)">Color</label>
  <InputText class="form-control" @bind-Value="CarInstance.Color" id="@nameof(Car.Color)" />
  <ValidationMessage For="() => CarInstance.Color"/>
</div>
<div>
  <label class="col-form-label" for="@nameof(Car.PetName)">Pet Name</label>
  <InputText class="form-control" @bind-Value="CarInstance.PetName" id="@nameof(Car.PetName)" />
  <ValidationMessage For="() => CarInstance.PetName"/>
</div>
<div>
  <label class="col-form-label" for="@nameof(Car.Price)">Price</label>
  <InputText class="form-control" @bind-Value="CarInstance.Price" id="@nameof(Car.Price)" />
  <ValidationMessage For="() => CarInstance.Price"/>
</div>
<div>
  <label class="col-form-label" for="@nameof(Car.DateBuilt)">Date Built</label>
  <InputDate class="form-control" @bind-Value="CarInstance.DateBuilt"
    id="@nameof(Car.DateBuilt)" />
  <ValidationMessage For="() => CarInstance.DateBuilt"/>
</div>
<div>
  <label class="col-form-label" for="@nameof(Car.IsDrivable)">Is Drivable</label>
  <InputCheckbox @bind-Value="CarInstance.IsDrivable" id="@nameof(Car.IsDrivable)" />
  <ValidationMessage For="() => CarInstance.IsDrivable"/>
</div>
@code {
  [Inject]
  private IMakeDataService MakeDataService { get; set; }
  [Parameter]
  [EditorRequired]
  public List<Make> Makes { get; set; }
  [Parameter]
  [EditorRequired]
  public Car CarInstance { get; set; } = default!;
}
```

Prompt: Add a new Razor component named CarList.razor to the CarHelpers folder. Clear out the contents and update it to the following:

```
<h3>Car List</h3>
<table class="table">
  <thead>
    <tr>
      <th>Make </th>
      <th>Color</th>
      <th>Pet Name</th>
      <th>Price</th>
      <th>DateBuilt</th>
      <th>Is Drivable</th>
      <th></th>
    </tr>
  </thead>
  <tbody>
  @foreach (var c in Cars)
  {
    <tr>
      <td>@c.MakeNavigation.Name</td>
      <td>@c.Color</td>
      <td>@c.PetName</td>
      <td>@c.Price</td>
      <td>@c.DateBuilt?.ToShortDateString()</td>
      <td>@c.IsDrivable</td>
      <td>
        <EditHelper Id="@c.Id" RouteStart="cars"/>
        <DetailsHelper Id="@c.Id" RouteStart="cars"/>
        <DeleteHelper Id="@c.Id" RouteStart="cars"/>
      </td>
    </tr>
  }
  </tbody>
</table>
@code {
  [Parameter][EditorRequired]
  public List<Car> Cars { get; set; }
}
```

Prompt: Add the following to the _Imports.razor file (don't remove any existing entries):

```
@using AutoLot.Blazor.Helpers.CarHelpers
```

Manual

Part 1: Add the Navigation Helpers

- Add a new directory named `Helpers` to the `AutoLot.Blazor` project. In this folder, add another new folder named `ItemHelpers`. In this folder, add a new Razor component named `CreateHelper.razor`. Clear out the contents and update it to the following:

```
<a href="/@RouteStart/create" class="text-success">
  Create <i class="fa-solid fa-plus"></i></a>

@code {
  [Parameter] public string RouteStart { get; set; }
}
```

- Add a new Razor component named `DeleteHelper.razor` to the `ItemHelpers` folder. Clear out the contents and update it to the following:

```
<a href="/@RouteStart/delete/@Id" class="text-danger">
  Delete <i class="fa-solid fa-trash"></i></a> |
@code {
  [Parameter] public int Id { get; set; }
  [Parameter] public string RouteStart { get; set; }
}
```

- Add a new Razor component named `DetailsHelper.razor` to the `ItemHelpers` folder. Clear out the contents and update it to the following:

```
<a href="/@RouteStart/details/@Id" class="text-info">
   Details <i class="fa-solid fa-info-circle"></i></a> |
@code {
  [Parameter] public int Id { get; set; }
  [Parameter] public string RouteStart { get; set; }
}
```

- Add a new Razor component named `EditHelper.razor` to the `ItemHelpers` folder. Clear out the contents and update it to the following:

```
<a href="/@RouteStart/edit/@Id" class="text-warning">
  Edit <i class="fa-solid fa-edit"></i></a> |
@code {
  [Parameter] public int Id { get; set; }
  [Parameter] public string RouteStart { get; set; }
}
```

- Add a new Razor component named `ListHelper.razor` to the `ItemHelpers` folder. Clear out the contents and update it to the following:

```
<a href="/@RouteStart/index" class="text-primary">
  Back to List <i class="fa-solid fa-list"></i></a>
@code {
  [Parameter] public string RouteStart { get; set; }
}
```

- Update the `_Imports.razor` file to include the new directory:

```
@using AutoLot.Blazor.Helpers
@using AutoLot.Blazor.Helpers.ItemHelpers
```

Part 2: Add the Car Helpers

- Add a new folder named `CarHelpers` under the `Helpers` folder. Add a new Razor component named `CarDetail.razor` to the `CarHelpers` folder. Clear out the contents and update it to the following:

```
<h1>Details for @CarInstance.PetName</h1>
<dl class="row">
  <dt class="col-sm-2">Make</dt>
  <dd class="col-sm-10">@CarInstance?.MakeNavigation?.Name</dd>
  <dt class="col-sm-2">Color</dt>
  <dd class="col-sm-10">@CarInstance?.Color</dd>
  <dt class="col-sm-2">Pet Name</dt>
  <dd class="col-sm-10">@CarInstance?.PetName</dd>
  <dt class="col-sm-2">Price</dt>
  <dd class="col-sm-10">@CarInstance?.Price</dd>
  <dt class="col-sm-2">Date Built</dt>
  <dd class="col-sm-10">@CarInstance?.DateBuilt?.ToShortDateString()</dd>
  <dt class="col-sm-2">Is Drivable</dt>
  <dd class="col-sm-10">@CarInstance?.IsDrivable</dd>
</dl>
@code {
  [Parameter]
  [EditorRequired]
  public Car CarInstance { get; set; } = default!;
}
```

- Add a new Razor component named `CarEdit.razor` to the `CarHelpers` folder. Clear out the contents and update it to the following:

```
<DataAnnotationsValidator/>
<ValidationSummary/>
<div>
  <label class="col-form-label" for="@nameof(Car.MakeId)">Make</label>
  <InputSelect class="form-control" @bind-Value="CarInstance.MakeId" id="@nameof(Car.MakeId)">
    @foreach (var item in Makes)
    {
      <option value="@item.Id">@item.Name</option>
    }
  </InputSelect>
  <ValidationMessage For="() => CarInstance.MakeId"/>
</div>
<div>
  <label class="col-form-label" for="@nameof(Car.Color)">Color</label>
  <InputText class="form-control" @bind-Value="CarInstance.Color" id="@nameof(Car.Color)" />
  <ValidationMessage For="() => CarInstance.Color"/>
</div>
<div>
  <label class="col-form-label" for="@nameof(Car.PetName)">Pet Name</label>
  <InputText class="form-control" @bind-Value="CarInstance.PetName" id="@nameof(Car.PetName)" />
  <ValidationMessage For="() => CarInstance.PetName"/>
</div>
<div>
  <label class="col-form-label" for="@nameof(Car.Price)">Price</label>
  <InputText class="form-control" @bind-Value="CarInstance.Price" id="@nameof(Car.Price)" />
  <ValidationMessage For="() => CarInstance.Price"/>
</div>
<div>
  <label class="col-form-label" for="@nameof(Car.DateBuilt)">Date Built</label>
  <InputDate class="form-control" @bind-Value="CarInstance.DateBuilt"
    id="@nameof(Car.DateBuilt)" />
  <ValidationMessage For="() => CarInstance.DateBuilt"/>
</div>
<div>
  <label class="col-form-label" for="@nameof(Car.IsDrivable)">Is Drivable</label>
  <InputCheckbox @bind-Value="CarInstance.IsDrivable" id="@nameof(Car.IsDrivable)" />
  <ValidationMessage For="() => CarInstance.IsDrivable"/>
</div>
@code {
  [Inject]
  private IMakeDataService MakeDataService { get; set; }
  [Parameter]
  [EditorRequired]
  public List<Make> Makes { get; set; }
  [Parameter]
  [EditorRequired]
  public Car CarInstance { get; set; } = default!;
}
```

- Add a new Razor component named `CarList.razor` to the `CarHelpers` folder. Clear out the contents and update it to the following:

```
<h3>Car List</h3>
<table class="table">
  <thead>
    <tr>
      <th>Make </th>
      <th>Color</th>
      <th>Pet Name</th>
      <th>Price</th>
      <th>DateBuilt</th>
      <th>Is Drivable</th>
      <th></th>
    </tr>
  </thead>
  <tbody>
  @foreach (var c in Cars)
  {
    <tr>
      <td>@c.MakeNavigation.Name</td>
      <td>@c.Color</td>
      <td>@c.PetName</td>
      <td>@c.Price</td>
      <td>@c.DateBuilt?.ToShortDateString()</td>
      <td>@c.IsDrivable</td>
      <td>
        <EditHelper Id="@c.Id" RouteStart="cars"/>
        <DetailsHelper Id="@c.Id" RouteStart="cars"/>
        <DeleteHelper Id="@c.Id" RouteStart="cars"/>
      </td>
    </tr>
  }
  </tbody>
</table>
@code {
  [Parameter][EditorRequired]
  public List<Car> Cars { get; set; }
}
```

- Update the `_Imports.razor` file to include the new directory:

```
@using AutoLot.Blazor.Helpers.CarHelpers
```

Summary

This lab added navigation and template helpers to the client application.

Next Steps

The following lab will finish the Car pages.