Build an EF and ASP.NET Core 2 App HOL

Welcome to the Build an Entity Framework Core and ASP.NET Core 2 Application in a Day Hands On Lab. This lab walks you through creating a View Component and cleans up some miscellaneous partial views.

Prior to starting this lab, you must have completed Lab 7.

All labs and files are available at https://github.com/skimedic/dotnetcore_hol.

Part 1: Create the View Component Server-Side Code

1) Create a new folder in the MVC project named ViewComponents.

2) Add a new class named Menu.cs.

3) Add the following using statements:

```
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.ViewComponents;
using SpyStore_HOL.DAL.Repos.Interfaces;
```

4) Update the code to the following:

```
public class Menu : ViewComponent
{
  private readonly ICategoryRepo _categoryRepo;
  public Menu(ICategoryRepo categoryRepo)
  {
    _categoryRepo = categoryRepo;
  }
  public async Task<IViewComponentResult> InvokeAsync()
  {
    var cats = _categoryRepo.GetAll();
    if (cats == null)
    {
      return new ContentViewComponentResult("There was an error getting the categories");
    }
    return View("MenuView", cats);
  }
}
```

Part 2: Update the ViewImports and Create the Partial View

Step 1: Update the _ViewImports.cshtml File

1) Open the _ViewImports.cshtml file under the Views folder.

2) Add the following using statements to the file:

```
@using SpyStore_HOL.Models.Entities
@using SpyStore_HOL.Models.ViewModels
@using SpyStore_HOL.Models.ViewModels.Base
@using System.Collections.Generic
@using Microsoft.AspNetCore.Mvc.Rendering
```

3) Add the following to the end of the file:

```
@addTagHelper *, SpyStore_HOL.MVC
```

Step 1: Create the MenuView partial view

1) Add a new folder named Components under the Views\Shared folder.

2) Add a new folder named Menu under the Components folder.

3) Add a new partial view named MenuView.cshtml in the new folder.

4) Update the code to match the following:

```
@model IEnumerable<Category>
<li><a asp-controller="Products" asp-action="Featured">Featured</a></li>
@foreach (var item in Model)
{
   <li><a asp-controller="Products"
       asp-action="ProductList"
       asp-route-id="@item.Id">@item.CategoryName</a></li>
}
```

Part 3: Add the LoginView partial view

The Login partial view leverages tag helpers, covered later.

1) Create a new partial view named LoginView.cshtml under the Views\Shared folder.

2) Add the following to the partial view:

```
<ul class="nav navbar-nav">
   <li><a asp-controller="Cart" asp-action="Index" asp-route-customerId="@ViewBag.CustomerId" title="Shopping
Cart"><span class="glyphicon glyphicon-shopping-cart"></span> Cart</a></li>
   <li><a asp-controller="Orders" asp-action="Index" asp-route-customerId="@ViewBag.CustomerId" title="Order
History"><span class="glyphicon glyphicon-tag"></span> Orders</a></li>
   <li>
     <a href="#" class="dropdown-toggle" data-toggle="dropdown"><span class="glyphicon glyphicon-search"></span>
SEARCH</a>
     <div class="dropdown-menu search-dropdown-menu">
       <form asp-controller="Products" asp-action="Search" class="navbar-form navbar-left" role="search">
         <div class="input-group">
           <label class="sr-only" for="searchString">Search</label>
           <input type="text" id="searchString" name="searchString" class="form-control" placeholder="SEARCH">
           <span class="input-group-btn">
             <button class="btn btn-default" type="submit"><span class="glyphicon glyphicon-
search"></span></button>
           </span>
         </div>
```

```
        </form>
      </div>
    </li>
</ul>
```

Part 4: Update the Main Layout

1) Open the _Layout.cshtml file in Views\Shared

2) Replace the menu items with the following:

```
<vc:menu></vc:menu>
```

3) Add the call to render the LoginView partial view to the menu div, as follows:

```
<div class="navbar-collapse collapse">
  <ul class="nav navbar-nav">
    <vc:menu></vc:menu>
  </ul>
  @Html.Partial("LoginView")
</div>
```

Summary

The lab created the View Component and its view as well as added the LoginPartial view.

Next steps

In the next part of this tutorial series, you will create a custom tag helper