1. Create a Class under Models Folder with name ToDoItem.cs

ToDoItem.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace Day4Demos.Models

{

public class ToDoItem

{

public int Id { get; set; }

public bool IsDone { get; set; }

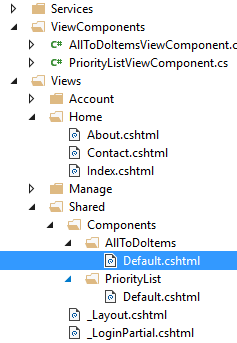
public int Priority { get; set; }

public string Name { get; set; }

}

}

1. Create Folder called ViewComponents



1. Create 2 Classes AllToDoItemsViewComponent and PriorityListView Component

AllToDoItemsViewComponet.cs

1. Add Class under the folder

using Day4Demos.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Rendering;

using Microsoft.EntityFrameworkCore;

namespace Day4Demos.ViewComponents

{

public class AllToDoItemsViewComponent : ViewComponent

{

public IViewComponentResult Invoke(int maxPriority, bool isDone)

{

var items = GetToDoItems();

return View(items);

}

private static List<ToDoItem> GetToDoItems()

{

List<ToDoItem> lstItems = new List<ToDoItem>();

for (int i = 0; i < 9; i++)

{

lstItems.Add(new ToDoItem()

{

IsDone = i % 3 == 0,

Name = "Task " + (i + 1),

Priority = i % 5 + 1

});

}

return lstItems;

}

}

}

1. Add Class Under Folder ViewComponents named PriorityListViewComponent

using Day4Demos.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Rendering;

using Microsoft.EntityFrameworkCore;

namespace Day4Demos.ViewComponents

{

public class PriorityListViewComponent : ViewComponent

{

public IViewComponentResult Invoke(int maxPriority, bool isDone)

{

var items = GetToDoItems().Where(x => x.IsDone == isDone &&

x.Priority <= maxPriority);

return View(items);

}

private static List<ToDoItem> GetToDoItems()

{

List<ToDoItem> lstItems = new List<ToDoItem>();

for (int i = 0; i < 9; i++)

{

lstItems.Add(new ToDoItem()

{

IsDone = i % 3 == 0,

Name = "Task " + (i + 1),

Priority = i % 5 + 1

});

}

return lstItems;

}

}

}

1. **AllToDoItems\Default.cshtml**

@model IEnumerable<Day4Demos.Models.ToDoItem>

<h3>All Items</h3>

<ul>

@foreach (var todo in Model)

{

<li>@todo.Name</li>

}

</ul>

1. **PriorityList\Default.cshtml**

@model IEnumerable<Day4Demos.Models.ToDoItem>

<h3>Priority Items</h3>

<ul>

@foreach (var todo in Model)

{

<li>@todo.Name</li>

}

</ul>

1. **Index.cshtml**

<div>

@await Component.InvokeAsync("PriorityList", new { maxPriority = 2, isDone = false })

</div>

<div>

@await Component.InvokeAsync("AllToDoItems")

</div>