

go deploy



Screenshots ▼

Module 5: Developing Views

Lab: Developing Views

? Scenario

To construct the user interface of a city's web application, your development team decided to add views. You have been asked to create the views to render a response to a browser.

Exercise 1: Adding Views to an MVC Application

? Scenario

To construct the user interface of a web application, views should be added to the web application. In this exercise you will add two views to the web application: Index and Details. The Index view will show a list of cities, and the Details view will show the details of a city.

The main tasks for this exercise are as follows:

- Add a view to show all the cities
- Run the application
- Add a view to show data for a city
- Add a Back link to a view
- Add a city name as a link to each city
- Run the application

Task 1: Add a view to show all the cities

- ☐ 1. Navigate to **D:\Allfiles\Mod05\Labfiles\01_CitiesWebsite_begin**, and then double-click **CitiesWebsite.sln**.

⚠ Note: If a **Security Warning for CitiesWebsite** dialog box appears, verify that the **Ask me for every project in this solution** check box is cleared, and then click **OK**.

- ☐ 2. In the **How do you want to open this file?** window, select **Visual Studio 2022** and ensure the checkbox for **Always use this app to open .sln files** is selected then click **OK**.
- ☐ 3. In the **CitiesWebsite - Microsoft Visual Studio** window, in Solution Explorer, expand **Services**, and then click **CityProvider.cs**.

go deploy



```
public CityProvider()  
{  
  
}
```

- ☐ 5. Place the cursor within the **CityProvider** constructor code block, and then type the following code:

```
_cities = CityInitializer();
```

- ☐ 6. In the **CitiesWebsite - Microsoft Visual Studio** window, in Solution Explorer, expand **Models**, and then click **City.cs**.

- ☐ 7. In the **City.cs** code window, locate the following code:

```
public City(string country, string cityName, string timeZone, CityPopulation popula  
{  
  
}
```

- ☐ 8. Place the cursor within the **City** constructor code block, and then type the following code:

```
Country = country;  
Name = cityName;  
TimeZone = timeZone;  
Population = population;
```

- ☐ 9. In the **CitiesWebsite - Microsoft Visual Studio** window, in Solution Explorer, under **Models**, click **CityPopulation.cs**.

- ☐ 10. In the **CityPopulation.cs** code window, locate the following code:

```
public CityPopulation(int year, int city, int urban, int metro)  
{  
  
}
```

- ☐ 11. Place the cursor within the **CityPopulation** constructor code block, and then type the following code:

```
Year = year;  
City = city;  
Urban = urban;  
Metro = metro;
```

- ☐ 12. In the **CitiesWebsite - Microsoft Visual Studio** window, in Solution Explorer, expand **Controllers**, and then click **CityController.cs**.

go deploy



```
public class CityController : Controller
{
```

- ☐ 14. Place the cursor at the end of the located code, press Enter, and then type the following code:

```
private ICityProvider _cities;
```

- ☐ 15. In the **CityController.cs** code window, select the following **CityController** constructor code:

```
public CityController()
{
}
```

- ☐ 16. Replace the selected code with the following code:

```
public CityController(ICityProvider cities)
{
    _cities = cities;
}
```

- ☐ 17. In the **CityController.cs** code window, locate the following code:

```
public IActionResult ShowCities()
{
```

- ☐ 18. Place the cursor after the { (opening brackets) sign, press Enter, and then type the following code:

```
ViewBag.Cities = _cities;
```

- ☐ 19. In the **CityController.cs** code window, right-click the following code, and then click **Add View...**

```
public IActionResult ShowCities()
```

- ☐ 20. In the **Add New Scaffolded Item** dialog box, ensure that the **Razor View** template (not the Empty template) is selected and click **Add**.

- ☐ 21. In the **Add Razor View** dialog box, ensure that the **View name** textbox contains the name **ShowCities**.

- ☐ 22. In the **Add Razor View** dialog box, ensure that both the **Create as a partial view** and **Use a layout page** check boxes are cleared, and then click **Add**.

- ☐ 23. In the **ShowCities.cshtml** code window, locate the following code:

go deploy



- ☐ 24. Place the cursor at the end of the located code, press Enter, and then type the following code:

```
<link rel="stylesheet" type="text/css" href="~/css/style.css" />
```

- ☐ 25. In **ShowCities.cshtml** code window, in the **BODY** element, type the following code:

```
<h1>Select City</h1>
@foreach (var item in ViewBag.Cities)
{
    <p>@item.Key</p>
}
```

Task 2: Run the application

- ☐ 1. In the **CitiesWebsite - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.
- ☐ 2. In the **CitiesWebsite - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Without Debugging**.

Note: The browser displays a **Select City** title and a list of cities below it: **Madrid, London, and Paris**.

- ☐ 1. In Microsoft Edge, click **Close**.

Task 3: Add a view to show data for a city

- ☐ 1. In the **CitiesWebsite - Microsoft Visual Studio** window, in Solution Explorer, under **Controllers**, click **CityController.cs**.
- ☐ 2. In the **CityController.cs** code window, select the following code:

```
public IActionResult ShowDataForCity()
```

- ☐ 3. Replace the selected code with the following code:

```
public IActionResult ShowDataForCity(string cityName)
```

- ☐ 4. Place the cursor after the { (opening brackets) sign, press Enter, and then type the following code:

```
ViewBag.City = _cities[cityName];
```

- ☐ 5. In the **CityController.cs** code window, right-click anywhere in the **ShowDataForCity** method, and then click **Add View...**

go deploy



- ☐ 7. In the **Add Razor View** dialog box, ensure that the value in the **Name:** textbox is **ShowDataForCity**.
- ☐ 8. In the **Add Razor View** dialog box, ensure that the **Create as a partial view** and **Use a layout page** check boxes are cleared, and then click **Add**.
- ☐ 9. In the **ShowDataForCity.cshtml** code window, locate the following code:

```
<title>ShowDataForCity</title>
```

- ☐ 10. Place the cursor at the end of the located code, press Enter, and then type the following code:

```
<link rel="stylesheet" type="text/css" href="~/css/style.css" />
```

- ☐ 11. In the **ShowDataForCity.cshtml** code window, in the **BODY** element, type the following code:

```
<div>
    <h2>@ViewBag.City.Name</h2>
    <p>Country: @ViewBag.City.Country</p>
    <p>Time zone: @ViewBag.City.TimeZone</p>
    <span>
```

- ☐ 12. In the **CitiesWebsite - Microsoft Visual Studio** window, in Solution Explorer, under **Controllers**, click **CityController.cs**.
- ☐ 13. In the **CityController.cs** code window, select the following code:

```
return Content(cityName);
```

- ☐ 14. Replace the selected code with the following code:

```
return File($"images\\{cityName}.jpg", "image/jpeg");
```

Task 4: Add links to the views by using tag helpers

- ☐ 1. In the **CitiesWebsite - Microsoft Visual Studio** window, in Solution Explorer, right-click **Views**, point to **Add**, and then click **New Item....**
- ☐ 2. In the **Add New Item -- CitiesWebsite** dialog box, choose **Razor View Imports**, and then click **Add**.
- ☐ 3. In the **_ViewImports.cshtml** code window, type the following code:

go deploy




- ☐ 4. In the **CitiesWebsite - Microsoft Visual Studio** window, in Solution Explorer, expand **Views**, expand **City**, and then click **ShowDataForCity.cshtml**.
- ☐ 5. In the **ShowDataForCity.cshtml** code window, locate the following code:

```
 <span></span>
```

- ☐ 6. Place the cursor at the end of the located code, press Enter, and then type the following code:

```
 <a asp-action="ShowCities">Back</a>
```

- ☐ 7. In the **CitiesWebsite - Microsoft Visual Studio** window, in Solution Explorer, under **Views**, under **City**, click **ShowCities.cshtml**.
- ☐ 8. In the **ShowCities.cshtml** code window, select the following code:

```
 <p>@item.Key</p>
```

- ☐ 9. Replace the selected code with the following code:

```
 <h2>  
    <a asp-action="ShowDataForCity" asp-route-cityname="@item.Key">@item.Key</a>  
</h2>
```

Task 5: Run the application

- ☐ 1. In the **CitiesWebsite - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.
- ☐ 2. In the **CitiesWebsite - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Without Debugging**.
- ☐ 3. In Microsoft Edge, click **London**.

 **Note:** The browser displays the city's name, details, mini map, and a **Back** link.

- ☐ 4. In Microsoft Edge, click **Back**.
- ☐ 5. In Microsoft Edge, click **Close**.

✓ **Result:** At the end of this exercise, you have added views to an MVC application, passed data from a controller to a view using **ViewBag**, and navigated between pages by using helpers.

go deploy



**? Scenario**


You have been asked to display the population of each city. To do this, you have been asked to add a partial view. In this exercise, you will create a partial view and embed it in the ShowDataForCity view.

The main tasks for this exercise are as follows:

- Add a partial view
- Use the partial view in the ShowDataForCity view
- Run the application

Task 1: Add a partial view

- ☐ 1. In the **CitiesWebsite - Microsoft Visual Studio** window, in Solution Explorer, right-click **Views**, point to **Add**, and then click **New Folder**.
- ☐ 2. In the **NewFolder** box, type  **Shared**, and then press Enter.
- ☐ 3. In Solution Explorer, right-click **Shared**, point to **Add**, and then click **View....**
- ☐ 4. In the **Add New Scaffolded Item** dialog box, ensure that the **Razor View** template (not the Empty template) is selected, and then click **Add**.
- ☐ 5. In the **Add Razor View** dialog box, in the **Name:** textbox, type  **_CityPopulation**.
- ☐ 6. In the **Add Razor View** dialog box, ensure that the **Create as a partial view** check box is selected, and then click **Add**.
- ☐ 7. In the **_CityPopulation.cshtml** code window, replace all the content with the following code:

```
 @inject CitiesWebsite.Services.ICityFormatter cityFormatter

@{
    CitiesWebsite.Models.CityPopulation population = ViewBag.City.Population;
}

<h3>City Population (@population.Year)</h3>
<p>City: @cityFormatter.GetFormattedPopulation(@population.City)</p>
<p>Urban: @cityFormatter.GetFormattedPopulation(@population.Urban)</p>
<p>Metro: @cityFormatter.GetFormattedPopulation(@population.Metro)</p>
```

Task 2: Use the partial view in the ShowDataForCity view

- ☐ 1. In the **CitiesWebsite - Microsoft Visual Studio** window, in Solution Explorer, under **Views**, under **City**, click **ShowDataForCity.cshtml**.

go deploy



```
<span></span>
```

- ☐ 3. Place the cursor at the end of the located code, press Enter, and then type the following code:

```
@await Html.PartialAsync("_CityPopulation")
```

Task 3: Run the application

- ☐ 1. In the **CitiesWebsite - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.
- ☐ 2. In the **CitiesWebsite - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Without Debugging**.
- ☐ 3. In Microsoft Edge, click **Madrid**.

Note: The browser displays the city's name, details, mini map, population, and a **Back** link.

- ☐ 4. In Microsoft Edge, click **Close**.

✓ **Result:** At the end of this exercise, you will have used partial views, and used services inside a view by using the **@inject** directive.

Exercise 3: Adding a View Component

? Scenario

Currently, in the ShowCities view, for each city, you show a link with the name of the city. You have been asked to show for each city in the ShowCities view, the country to which the city belongs and a mini map of the city. To implement this you have been asked to use a view component. In this exercise, you will create a view component and embed it in the ShowCities view.

The main tasks for this exercise are as follows:



- Add a view component class
- Add a view component view
- Use the view component
- Run the application

Task 1: Add a view component class

- ☐ 1. In the **CitiesWebsite - Microsoft Visual Studio** window, in Solution Explorer, right-click

go deploy




- ☐ 2. In the **NewFolder** box, type  **ViewComponents**, and then press Enter.
- ☐ 3. In the **CitiesWebsite - Microsoft Visual Studio** window, in Solution Explorer, right-click **ViewComponents**, point to **Add**, and then click **Class....**
- ☐ 4. In the **Add New Item -- CitiesWebsite** dialog box, in the **Name:** box, type  **CityViewComponent**, and then click **Add**.
- ☐ 5. In the **CityViewComponent.cs** code window, select the following code:

```
 public class CityViewComponent
```

- ☐ 6. Add a base class to the **CityViewComponent** class as follows:

```
 public class CityViewComponent : ViewComponent
```

- ☐ 7. In the **CityViewComponent.cs** code window, place the cursor within the **CityViewComponent** code block, and then type the following code:


```
 private ICityProvider _cities;

public CityViewComponent(ICityProvider cities)
{
    _cities = cities;
}

public async Task<IViewComponentResult> InvokeAsync(string cityName)
{
    ViewBag.CurrentCity = await GetCity(cityName);
    return View("SelectCity");
}


private Task<City> GetCity(string cityName)
{
    return Task.FromResult<City>(_cities[cityName]);
}
```

Task 2: Add a view component view

- ☐ 1. In the **CitiesWebsite - Microsoft Visual Studio** window, in Solution Explorer, under **Views**, right-click **Shared**, point to **Add**, and then click **New Folder**.
- ☐ 2. In the **NewFolder** box, type  **Components**, and then press Enter.
- ☐ 3. In the **CitiesWebsite - Microsoft Visual Studio** window, in Solution Explorer, right-click **Components**, point to **Add**, and then click **New Folder**.

go deploy



- ☐ 5. In the **CitiesWebsite - Microsoft Visual Studio** window, in Solution Explorer, right-click the **City** folder you just created, point to **Add**, and then click **View...**
- ☐ 6. In the **Add New Scaffolded Item** dialog box, ensure that the **Razor View** template (not the Empty template) is selected, and then click **Add**.
- ☐ 7. In the **Add Razor View** dialog box, in the **Name:** textbox, type  **SelectCity**.
- ☐ 8. In the **Add Razor View** dialog box, ensure that the **Create as a partial view** check box is selected, and then click **Add**.
- ☐ 9. In the **SelectCity.cshtml** code window, replace all the content with the following code:


```
 <div>
    <h2>
        <a asp-action="ShowDataForCity" asp-route-cityname=@ViewBag.CurrentCity.Name>
    </h2>
    
```

Task 3: Use the view component

- ☐ 1. In the **CitiesWebsite - Microsoft Visual Studio** window, in Solution Explorer, under **Views**, under **City**, click **ShowCities.cshtml**.
- ☐ 2. In the **ShowCities.cshtml** code window, select the following code:


```
 <h2>
    <a asp-action="ShowDataForCity" asp-route-cityname="@item.Key">@item.Key</a>
</h2>
```

- ☐ 3. Replace the selected code with the following code:

```
 @await Component.InvokeAsync("City", item.Key)
```

Task 4: Run the application

- ☐ 1. In the **CitiesWebsite - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.
- ☐ 2. In the **CitiesWebsite - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Without Debugging**.

 **Note:** The browser displays a list of cities. Each has a header link and mini map image of the region.

- ☐ 3. In Microsoft Edge, click **Madrid (Capital of Spain)**.

go deploy



- 5. In the **Citieswebsite -- Microsoft Visual Studio** window, on the **File** menu, click **Exit**.

✓ **Result:** At the end of this exercise, you have created view components, and embedded them in a view.