

Walkthrough 3 - MVC Introduction

Setup

This part of the walkthrough will continue the MVC introduction.



1. Open IntroMVC from the end of the previous walkthrough.

OtherController.cs

1. Add another Controller. Right-click the Controller folder, select Add / Controller..., choose the MVC Controller - Empty template, click Add. Name it **OtherController**.
2. Put the cursor in the Index method. Right-click and select Add View..., select Razor View - Empty, click Add. Accept the default name **Index.cshtml**, click Add.
3. A folder named Views will be created. Inside the Views folder, another folder named Other will be created. This is where Index.cshtml will reside.

Index.cshtml

1. Delete the existing code.
2.

 For more information on enabling MVC for empty projects, visit <https://go.microsoft.com/fwlink/?LinkID=397860>



3. Add a heading.
4.

`<h1>Hello from OtherController / Index</h1>`





OtherController.cs

1. Add a Post method that accepts a string and passes it to the view and then returns the view.
2.

```
public IActionResult Post(string id)
{
    ViewBag.id = id;
    return View();
}
```
3. Put the cursor in the Post method. Right-click and select Add View..., select Razor View - Empty, click Add. Name the view **Post.cshtml**, click Add.

Post.cshtml

1. Delete the existing code.
2.

 For more information on enabling MVC for empty projects, visit <https://go.microsoft.com/fwlink/?LinkID=397860>



3. Add two headings.
4.

`<h1>Hello from OtherController / Post</h1>
<h2>id=@ViewBag.id</h2>`
5. Press Ctrl+F5 to launch the site.
6. Navigate to the following URLs and note the results.
 - **http://localhost:12345/other/post/1**
 - **http://localhost:12345/other/post/**
 - **http://localhost:12345/other/**
 - **http://localhost:12345/**
 - **http://localhost:12345/other/post/abc**
 - **http://localhost:12345/other/post?id=def**
 - **http://localhost:12345/other/post?parm=def**

OtherController.cs

1. Change the id to an integer.
2.

```
public IActionResult Post(int id)
{
    ViewBag.id = id;
    return View();
}
```

- Save the file and refresh the browser. Note id is now 0, because that is the default value for an integer.
- Change the URL to **http://localhost:12345/other/post/10** and note the output.
- Change the id to a boolean.

```
6. public IActionResult Post(int id)
{
    ViewBag.id = id;
    return View();
}
```

- Save the file and refresh the browser. Note id is now False, because that is the default value for a boolean.
- Change the URL to **http://localhost:12345/other/post/true** and note the output.
- Change the id to be a nullable integer. This makes the default value null instead of 0.

```
10. public IActionResult Post(int? id)
{
    ViewBag.id = id;
    return View();
}
```

- Save the file and refresh the browser, note the output.
- Change the URL to **http://localhost:12345/other/post/10** and note the output.
- Change the id to have a default value.

```
14. public IActionResult Post(int id = -1)
{
    ViewBag.id = id;
    return View();
}
```

- Save the file and refresh the browser, note the output.
- Change the URL to **http://localhost:12345/other/post/abc** and note the output.
- Customize the route that the Post method will respond to.

```
18. [Route("stuff")]
public IActionResult Post(int id = -1)
{
    ViewBag.id = id;
    return View();
}
```

- Save the file and refresh the browser, the page won't be found because it's route is now different.
- Change the URL to **http://localhost:12345/stuff** and Post will respond.
- Customize the route that the Post method will use to look for a year, a month and string value name key.

```
22. [Route("stuff/{year:int}/{month:int}/{key}")]
public IActionResult Post(int id = -1, int year, int month, string key)
{
    ViewBag.id = id;
    ViewBag.year = year;
    ViewData["MONTH"] = month;
    ViewBag.key = key;
    return View();
}
```

- Save the file.

Post.cshtml

- Update the view to use the new data.

```
2. <h1>Hello from OtherController / Post</h1>
<h2>id=@ViewBag.id</h2>
<h2>year=@ViewBag.year</h2>
<h2>month=@ViewBag.month</h2>
<h2>key=@ViewData["Key"]</h2>
```

- Save the file and refresh the browser, the page won't be found because it's route is now different.
- Change the URL to **http://localhost:12345/stuff/2021/10/abc** and note the output.

OtherController.cs

- Add route constraints to ensure recent years and possible months.

```
2. [Route("stuff/{year:intmin(2019)}/{month:intrange(1,12)}/{key}")]
public IActionResult Post(int year, int month, string key)
{
    ViewBag.year = year;
    ViewData["MONTH"] = month;
    ViewBag.key = key;
    return View();
}
```

- Save the file and refresh the browser, output doesn't change.
- Change the URL to **http://localhost:12345/stuff/2021/99/abc**, the routing won't be satisfied and controller won't respond.
- Change the URL back to **http://localhost:12345/stuff/2021/10/abc**.

Post.cshtml

- Add some Razor code to show the current date/time and a week ago.

2.

```
<h1>Hello from OtherController / Post</h1>
<h2>year=@ViewBag.year</h2>
<h2>month=@ViewBag.month</h2>
<h2>key=@ViewData["Key"]</h2>
<p>Today: @DateTime.Now</p>
<p>Week ago: @(DateTime.Now - TimeSpan.FromDays(7))</p>
```

3. Save the file and refresh the browser, note the output.

4. Add a Razor code block.

5.

```
<h1>Hello from OtherController / Post</h1>
<h2>year=@ViewBag.year</h2>
<h2>month=@ViewBag.month</h2>
<h2>key=@ViewData["Key"]</h2>
<p>Today: @DateTime.Now</p>
<p>Week ago: @(DateTime.Now - TimeSpan.FromDays(7))</p>
@{
    var name = "Bob Loblaw";
    var len = name.Length;
}
<p>The length of @name is @len bytes.</p>
```

6. Save the file and refresh the browser, note the output.

7. Add a list using Razor.

8.

```
<h1>Hello from OtherController / Post</h1>
<h2>year=@ViewBag.year</h2>
<h2>month=@ViewBag.month</h2>
<h2>key=@ViewData["Key"]</h2>
<p>Today: @DateTime.Now</p>
<p>Week ago: @(DateTime.Now - TimeSpan.FromDays(7))</p>
@{
    var name = "Bob Loblaw";
    var len = name.Length;
}
<p>The length of @name is @len bytes.</p>
<ul>
    @for (int i = 0; i < ViewBag.month; i++)
    {
        <li>Month @i</li>
    }
</ul>
```

9. Save the file and refresh the browser, note the output.

10. Month 0 doesn't really make sense, update the code.

11.

```
<h1>Hello from OtherController / Post</h1>
<h2>year=@ViewBag.year</h2>
<h2>month=@ViewBag.month</h2>
<h2>key=@ViewData["Key"]</h2>
<p>Today: @DateTime.Now</p>
<p>Week ago: @(DateTime.Now - TimeSpan.FromDays(7))</p>
@{
    var name = "Bob Loblaw";
    var len = name.Length;
}
<p>The length of @name is @len bytes.</p>
<ul>
    @for (int i = 0; i < ViewBag.month; i++)
    {
        <li>Month @(i + 1)</li>
    }
</ul>
```

12. Save the file and refresh the browser, note the output.

13. Add some more Razor code to display the month name.

14.

```
<h1>Hello from OtherController / Post</h1>
<h2>year=@ViewBag.year</h2>
<h2>month=@ViewBag.month</h2>
<h2>key=@ViewData["Key"]</h2>
<p>Today: @DateTime.Now</p>
<p>Week ago: @(DateTime.Now - TimeSpan.FromDays(7))</p>
@{
    var name = "Bob Loblaw";
    var len = name.Length;
}
<p>The length of @name is @len bytes.</p>
<ul>
    @for (int i = 0; i < ViewBag.month; i++)
    {
        <li>Month @(i + 1) is @(new DateTime(2021, i + 1, 1).ToString("MMMM"))</li>
    }
</ul>
```

15. Save the file and refresh the browser, note the output.

HomeController.cs

1. Revert the Index method to return a view.

2.

```
public IActionResult Index()
{
    return new ContentResult { Content = "Hello from HomeController / Index." }View();
}
```

3. Save the file and change the URL to **http://localhost:12345/**, an exception will be thrown, examine it.

Index.cshtml

1. Create the **Home** folder under the Views folder.
2. Move the index.html file from the wwwroot folder into the new Views / Home folder by dragging it. Rename the file **Index.cshtml**.
3. Refresh the browser, the page should appear.
4. In the footer, near the bottom of the page, replace the JavaScript year with some Razor code.

5.

```
...
    <footer class="border-top footer text-muted">
      <div class="container">
        &copy; <script>@DateTime.Now.Year>document.write(new Date().getFullYear())</script>@DateTime.Now.Year - MVC Intro
      </div>
    </footer>
  </body>
</html>
```

6. Save the file and refresh the browser. Nothing will look different, but the year isn't generated with JavaScript anymore.
7. In the navbar-collapse div, update the Page Two link to link to Index action of the Other controller.

8.

```
...
<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
  <ul class="navbar-nav flex-grow-1">
    <li class="nav-item">
      <a class="nav-link text-dark" href="/pageone.html">Page One</a>
    </li>
    <li class="nav-item">
      <a class="nav-link text-dark" href="/pagetwo.html@Url.Action("Index", "Other")">Page TwoOther</a>
    </li>
  </ul>
</div>
...
```

9. Save the file and refresh the browser. Try the updated link. Click the browser back button. Click the Page One link, note that it's menu is the old menu. Click the MVC Intro link.

_Layout.cshtml

1. Create the **Shared** folder under the Views folder.
2. Copy Views / Home / Index.cshtml file into the new Shared folder and rename it **_Layout.cshtml**.
3. Update the title tag to use ViewData, so that each page can have a customized title.

4.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Home Page@ViewData["Title"] - MVC Intro</title>
  <link rel="stylesheet" href="/css/bootstrap.min.css" />
  <link rel="stylesheet" href="/css/site.css" />
</head>
...
```

5. In the main section, delete the div that displays the welcome message and replace it with a call to the RenderBody method.

6.

```
...
  <div class="container">
    <main role="main" class="pb-3">
      <div class="text-center">
        <h1 class="display-4">Welcome to MVC</h1>
      </div>
      @RenderBody()
    </main>
  </div>

  <footer class="border-top footer text-muted">
    <div class="container">
      &copy; @DateTime.Now.Year - MVC Intro
    </div>
  </footer>
</body>
</html>
```

7. Save the file.

Index.cshtml

1. Open Views / Home / Index.cshtml.
2. Delete all of the code and re-create the header.

3.

```
<div class="text-center">
  <h1 class="display-4">Welcome to MVC</h1>
</div>
```

4. Add a Razor code block to specify the Layout and set the page title.

5.

```
@{
  Layout = "_Layout";
  ViewData["Title"] = "Home Page";
}
<div class="text-center">
  <h1 class="display-4">Welcome to MVC</h1>
</div>
```

6. Save the file and refresh the browser, page should look the same.

IntroMVC.csproj

1. Sometimes, after the Index.cshtml manipulation, the _Layout.cshtml isn't found.
2. If this error occurs, click the project in the Solution Explorer and the IntroMVC.csproj file will open. Delete any ItemGroup tags.

```
3. <Project Sdk="Microsoft.NET.Sdk.Web">

  <PropertyGroup>
    <TargetFramework>net5.0</TargetFramework>
  </PropertyGroup>

  <ItemGroup>
    <Content Remove="Views\Shared\_Layout.cshtml" />
  </ItemGroup>

  <ItemGroup>
    <None Include="Views\Home\Index.cshtml" />
    <None Include="Views\Shared\_Layout.cshtml" />
  </ItemGroup>

</Project>
```

4. Save the file and refresh the browser, page should look the same.

Index.cshtml

1. Open Views / Other / Index.cshtml.
2. Add a Razor code block to specify the Layout and set the page title.

```
3. @{
    Layout = "_Layout";
    ViewData["Title"] = "Other Page";
}
<h1>Hello from OtherController / Index</h1>
```

4. Save the file.
5. In the browser, click the Other link, the new view is presented.

OtherController.cs

1. Add a new method to serve page one.

```
2. public IActionResult PageOne()
{
    return View();
}
```

3. Save the file.
4. Put the cursor in the PageOne method. Right-click and select Add View..., select Razor View - Empty, click Add. Name the view **PageOne.cshtml**, click Add.

PageOne.cshtml

1. Delete the existing code.

```
2. @*
For more information on enabling MVC for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860
*@
@{
}
```

3. Add a Razor code block to specify the Layout and set the page title. Also add the same heading that was in pageone.html.

```
4. @{
    Layout = "_Layout";
    ViewData["Title"] = "Page One";
}
<div class="text-center">
    <h1 class="display-4">This is page one</h1>
</div>
```

5. Save the file.

_Layout.cshtml

1. In the navbar-collapse div, update the Page One link to link to the PageOne action of the Other controller.

```
2. ...
<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
  <ul class="navbar-nav flex-grow-1">
    <li class="nav-item">
      <a class="nav-link text-dark" href="/pageone.html@Url.Action("PageOne", "Other")">Page One</a>
    </li>
    <li class="nav-item">
      <a class="nav-link text-dark" href="@Url.Action("Index", "Other")">Other</a>
    </li>
  </ul>
</div>
...
```

- 3. Save the file.
- 4. Refresh the browser; the menu should now work on all pages. Click the Page One link.

pageone.html

- 1. Delete wwwroot / pageone.html as it is no longer needed.

Person.cs

- 1. Create the **Models** folder under the project.
- 2. Right-click the Models folder, select Add / Class... and name it **Person.cs**.

3.

```
public class Person
{
    public int Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public DateTime DateOfBirth { get; set; }
}
```

- 4. Save the file.

OtherController.cs

- 1. In the PageOne method, instantiate a new Person object, use quick actions to add the **using IntroMVC.Models;** directive. Pass the person to the View.

2.

```
public IActionResult PageOne()
{
    var person = new Person
    {
        Id = 1,
        FirstName = "Bob",
        LastName = "Loblaw",
        DateOfBirth = DateTime.Now.AddDays(-10000)
    };
    return View(person);
}
```

- 3. Save the file.

PageOne.cshtml

- 1. Add the model to the view. Display the person.

2.

```
@model IntroMVC.Models.Person
@{
    Layout = "_Layout";
    ViewData["Title"] = "Page One";
}
<div class="text-center">
    <h1 class="display-4">This is page one</h1>
</div>
<h2>Person</h2>
<dl>
    <dt>Id</dt>
    <dd>@Model.Id</dd>
    <dt>First name</dt>
    <dd>@Model.FirstName</dd>
    <dt>Last name</dt>
    <dd>@Model.LastName</dd>
    <dt>Date of birth</dt>
    <dd>@Model.DateOfBirth</dd>
</dl>
```

- 3. Save the file, refresh the browser.