

Walkthrough 17 - Razor Pages Introduction

Setup

This lab will explore Razor Pages.

1. Start Visual Studio.
2. Click Create a new project.
3. Set language to **C#** and project type to **Web**.
4. Select the ASP.NET Core Web App template, click Next.
5. Set Project name to **RazorIntro**.
6. Set Location to a **folder of your choosing**.
7. Ensure Place solution and project in the same directory is not selected, click Next.
8. Set version to **.NET 5.0**, unselect Configure for HTTPS, click Create.

Index.cshtml

1. Open Pages / Index.cshtml.
2. Notice the @model property is IndexModel.

Index.cshtml.cs

1. Open Pages / Index.cshtml / Index.cshtml.cs.
2. Notice the class is IndexModel which inherits from PageModel.
3. Put the cursor on PageModel and press F1. Explore the help page.
4. Notice the OnGet method. This method returns the Index.cshtml page.
5. Add a property to the class.

6.

```
public class IndexModel : PageModel
{
    private readonly ILogger<IndexModel> _logger;
    public string FirstName { get; set; }

    public IndexModel(ILogger<IndexModel> logger)
    {
        _logger = logger;
    }

    public void OnGet()
    {

    }
}
```

Index.cshtml

1. Display the property on the Index page.

2.

```
@page
@model IndexModel
@{
    ViewData["Title"] = "Home page";
}

<div class="text-center">
    <h1 class="display-4">Welcome @Model.FirstName</h1>
    <p>Learn about <a href="https://docs.microsoft.com/aspnet/core">building Web apps with ASP.NET Core</a>.</p>
</div>
```

Index.cshtml.cs

1. At the moment, the FirstName property can only be displayed on the Index page. To write to the property, it has to be marked as writable.

2.

```
public class IndexModel : PageModel
{
    private readonly ILogger<IndexModel> _logger;
    [BindProperty]
    public string FirstName { get; set; }

    public IndexModel(ILogger<IndexModel> logger)
    {
        _logger = logger;
    }

    public void OnGet()
    {

    }
}
```

3. The FirstName property could now be written to on a Post operation. In order to allow it to be specified in a URL (which would be a Get operation), it must be further annotated.

```
4. public class IndexModel : PageModel
{
    private readonly ILogger<IndexModel> _logger;
    [BindProperty(SupportsGet = true)]
    public string FirstName { get; set; }

    public IndexModel(ILogger<IndexModel> logger)
    {
        _logger = logger;
    }

    public void OnGet()
    {
    }
}
```

- 5. Run the site. No name appears.
- 6. Change the URL to **http://localhost:12345/?FirstName=Bob**. The name appears.
- 7. Provide a default value for FirstName in OnGet.

```
8. public void OnGet()
{
    if (string.IsNullOrEmpty(FirstName))
    {
        FirstName = "User";
    }
}
```

- 9. Run the site, initially User will show up. Alter the URL to provide the FirstName parameter and it will appear again.
- 10. This technique is only appropriate for non-sensitive data, typically identifiers.

Index.cshtml

- 1. Update the page directive to optionally accept FirstName as route data.

```
2. @page "{FirstName?}"
@model IndexModel
...
```

- 3. Save the file.
- 4. Change the URL to **http://localhost:12345/Betty**. The name appears.

AddCourse.cshtml

- 1. Right-click the Pages folder and select Add / New Folder. Name the folder **Forms**.
- 2. Right-click the Forms folder and select Add / Razor Page... . Choose the Razor Page - Empty template, click Add.
- 3. Set Razor Page name to **AddCourse**, click Add.
- 4. Add a title and an h1 heading.

```
5. @page
@model RazorIntro.Pages.Forms.AddCourseModel
@{
    ViewBag.Title = "Add Course";
}

<h1>@ViewBag.Title</h1>
```

- 6. Save the file.

Course.cs

- 1. Add a **Models** folder to the project.
- 2. Add a class to the Models folder named **Course**. Add the **using System.ComponentModel.DataAnnotations;** directive.

```
3. public class Course
{
    [Display(Name = "Course Code")]
    public string CourseCode { get; set; }

    [Display(Name = "Course Name")]
    public string CourseName { get; set; }

    public int? Hours { get; set; }
}
```

- 4. Save the file.

AddCourse.cshtml.cs

- 1. Add Course as a property and add the **using RazorIntro.Models;** directive.

```
2. public class AddCourseModel : PageModel
{
    [BindProperty]
    public Course Course { get; set; }

    public void OnGet()
    {
    }
}
```

```
}  
}
```

AddCourse.cshtml

1. Add a form with text boxes for the 3 properties and a submit button.

```
2. @page  
@model RazorIntro.Pages.Forms.AddCourseModel  
@{  
    ViewBag.Title = "Add Course";  
}  
  
<h1>@ViewBag.Title</h1>  
  
<div class="col-md-4">  
    <form method="post">  
        <div class="form-group">  
            <label asp-for="Course.CourseCode" class="col-form-label"></label>  
            <input asp-for="Course.CourseCode" class="form-control" autofocus />  
        </div>  
  
        <div class="form-group">  
            <label asp-for="Course.CourseName" class="col-form-label"></label>  
            <input asp-for="Course.CourseName" class="form-control" />  
        </div>  
  
        <div class="form-group">  
            <label asp-for="Course.Hours" class="col-form-label"></label>  
            <input asp-for="Course.Hours" class="form-control" />  
        </div>  
  
        <button type="submit" class="btn btn-primary">Submit</button>  
    </form>  
</div>
```

- 3. Save the file.
- 4. Navigate to **http://localhost:12345/forms/addcourse** to see the new page.
- 5. Close the browser.

AddCourse.cshtml.cs

1. Create a Post method

```
2. public IActionResult OnPost()  
{  
    if (!ModelState.IsValid)  
    {  
        return Page();  
    }  
  
    // Persist the course to the data store  
  
    return RedirectToPage("/Index");  
}
```

_Layout.cshtml

- 1. Open Pages / Shared / _Layout.cshtml.
- 2. Add a menu item for the new page.

```
3. ...  
<li class="nav-item">  
    <a class="nav-link text-dark" asp-area="" asp-page="/Index">Home</a>  
</li>  
<li class="nav-item">  
    <a class="nav-link text-dark" asp-area="" asp-page="/Forms/AddCourse">Add Course</a>  
</li>  
<li class="nav-item">  
    <a class="nav-link text-dark" asp-area="" asp-page="/Privacy">Privacy</a>  
</li>  
...
```

AddCourse.cshtml.cs

- 1. Add a breakpoint to the if (!ModelState.IsValid) line.
- 2. Run the site in debug mode.
- 3. Add a course.
- 4. When the debugger stops, hover over the Course property to see it is populated.
- 5. Allow the program to continue.
- 6. Stop the program, remove the breakpoint.

Index.cshtml.cs

1. Change the FirstName property to **CourseName**.

```
2. public class IndexModel : PageModel  
{  
    private readonly ILogger<IndexModel> _logger;  
    [BindProperty(SupportsGet = true)]  
    public string FirstNameCourseName { get; set; }  
}
```

```
public IndexModel(ILogger<IndexModel> logger)
{
    _logger = logger;
}

public void OnGet()
{
    if (string.IsNullOrEmpty(FirstNameCourseName))
    {
        FirstNameCourseName = "UserMohawk College";
    }
}
}
```

3. Save the file.

Index.chtml

1. Update the view to use CourseName.

```
2. @page "{FirstNameCourseName?}"
@model IndexModel
@{
    ViewData["Title"] = "Home page";
}

<div class="text-center">
    <h1 class="display-4">Welcome to @Model.FirstNameCourseName</h1>
    <p>Learn about <a href="https://docs.microsoft.com/aspnet/core">building Web apps with ASP.NET Core</a>.</p>
</div>
```

3. Save the file.

4. Run the site, the default greeting is presented.

AddCourse.cshtml.cs

1. Add a route value to the redirect as an anonymous object.

```
2. public IActionResult OnPost()
{
    if (ModelState.IsValid == false)
    {
        return Page();
    }

    // Persist the course to the data store

    return RedirectToPage("/Index", new { CourseName = Course.CourseName });
}
```

3. Since the property name of the parameter is the same as the property name, it can be removed.

```
4. public IActionResult OnPost()
{
    if (ModelState.IsValid == false)
    {
        return Page();
    }

    // Persist the course to the data store

    return RedirectToPage("/Index", new { CourseName = Course.CourseName });
}
```

5. Run the site, add a new course.