

Thực hành (2) CRUD trong ASP.NET Core MVC

Hướng dẫn tự học lập trình ASP.NET Core toàn tập > Thực hành (2) CRUD trong ASP.NET Core MVC

CRUD (Create-Retrieve-Update-Delete) là nhóm chức năng cơ bản trên dữ liệu mà mọi ứng dụng đều phải có. Trong ASP.NET Core MVC nhóm chức năng này được thực hiện một cách dễ dàng. ASP.NET Core MVC thậm chí còn cung cấp khả năng scaffolding – tự động sinh code controller và view dựa trên thông tin của model.

Bài học này nối tiếp bài thực hành (1) trình bày cách xây dựng một ứng dụng quản lý đơn giản. Trong bài học này bạn sẽ tiếp tục thực hiện chức năng Delete – Update – Create.

NỘI DUNG CỦA BÀI [Ấn]

1. Xóa dữ liệu
2. Cập nhật dữ liệu
3. Thêm dữ liệu mới
4. Tái sử dụng code với partial view
5. Kết luận

Xóa dữ liệu

Chức năng xóa dữ liệu thực hiện theo quy trình sau:

1. Chọn lệnh xóa (từ Index hoặc Details view);
2. Hiển thị thông tin chi tiết của dữ liệu và yêu cầu xác nhận xóa;
3. Nếu người dùng xác nhận xóa => xóa dữ liệu và điều hướng về trang Index;
4. Nếu người dùng hủy xóa => điều hướng về trang Index.

Bước 1. Điều chỉnh BookController

Bổ sung hai action sau vào BookController:

```
1. public IActionResult Delete(int id) {
2.     var b = _service.Get(id);
3.     if (b == null) return NotFound();
4.     else return View(b);
5. }
6.
7. [HttpPost]
8. public IActionResult Delete(Book book) {
9.     _service.Delete(book.Id);
10.    _service.SaveChanges();
11.    return RedirectToAction("Index");
12. }
```

Hai action này tuân theo một mẫu xử lý phổ biến trong ASP.NET Core: Get – Post – Redirect.

Trong chức năng xóa, mẫu này thể hiện như sau:

1. Người dùng thông qua link gọi tới action Delete lần đầu tiên với truy vấn GET, phương thức Delete(int id) sẽ được gọi.
2. Action Delete(int id) lựa chọn view tương ứng. Trong view này phải có một form gọi trở về action Delete với truy vấn POST.
3. Ở lần gọi action Delete thứ hai từ form với truy vấn POST, action Delete(Book book) sẽ được gọi. Về kỹ thuật để yêu cầu kích hoạt action theo truy vấn POST, chúng ta đặt attribute [HttpPost] trước action đó.
4. Khi action Delete(Book book) thực hiện xong người dùng sẽ được điều hướng về trang Index. Việc điều hướng này giúp tránh tình trạng post dữ liệu nhiều lần – vốn có thể xem như một vấn đề an ninh.

Chú ý rằng [HttpPost] là bắt buộc đối với action Delete(Book book). ASP.NET Core MVC chỉ phân biệt các action theo tên gọi chứ không quan tâm đến danh sách tham số. Do vậy, Delete(int id) và Delete(Book book) trong ASP.NET Core MVC là như nhau.

ASP.NET Core phân biệt phương thức action dựa trên loại truy vấn mà nó xử lý thông qua attribute [Http...], bao gồm [HttpPost], [HttpGet], v.v..

Mẫu Get – Post – Redirect được sử dụng cho cả 3 loại chức năng Delete, Update và Create.

Bước 2. Xây dựng Delete view

Sử dụng scaffolding hoặc tự mình tạo Delete view (/Views/Book/Delete.cshtml) với code như sau:

```
1. @model BookMan.Mvc.Models.Book
2.
3. @{
4.     ViewData["Title"] = "Confirmation";
5. }
6.
7. <p class="p-3 alert-danger">Are you sure you want to delete this?</p>
8.
9. <div>
10.     <p class="h3">@Model.Name</p>
11.     <hr />
12.     <dl class="row">
13.         <dt class="col-sm-2"><label asp-for="Name"></label></dt>
14.         <dd class="col-sm-10">@Model.Name</dd>
15.         <dt class="col-sm-2"><label asp-for="Authors"></label></dt>
16.         <dd class="col-sm-10">@Model.Authors</dd>
17.         <dt class="col-sm-2"><label asp-for="Publisher"></label></dt>
18.         <dd class="col-sm-10">@Model.Publisher</dd>
19.         <dt class="col-sm-2"><label asp-for="Year"></label></dt>
20.         <dd class="col-sm-10">@Model.Year</dd>
21.         <dt class="col-sm-2"><label asp-for="DataFile"></label></dt>
22.         <dd class="col-sm-10">@Model.DataFile</dd>
23.         <dt class="col-sm-2"><label asp-for="Description"></label></dt>
24.         <dd class="col-sm-10">@Model.Description</dd>
25.     </dl>
26. </div>
27.
28. <hr />
29.
30. <form asp-action="Delete">
31.     <input type="hidden" asp-for="Id" />
32.     <input type="submit" value="Delete" class="btn btn-outline-danger btn-sm" /> |
33.     <a asp-action="Index" class="btn btn-sm btn-outline-secondary">Back to List</a>
34. </form>
```

Bước 3. Thêm nút lệnh Delete vào Index và Details view

Để kích hoạt chức năng xóa dữ liệu, bổ sung thẻ sau vào Index view (vào cạnh nút Details):

```
1. <a asp-action="Delete" asp-route-id="@item.Id" class="btn btn-sm btn-outline-danger">De
```

Tương tự, bổ sung thẻ sau vào Details view (vào cạnh nút Back to List):

```
1. <a asp-action="Delete" asp-route-id="@Model.Id" class="btn btn-sm btn-outline-danger">De
```

Bước 4. Chạy thử ứng dụng

Kiểm tra xem bạn đã tạo thư mục Data trong project hay chưa. Nếu chưa, hãy tạo thư mục Data trực thuộc dự án và chạy chương trình.

Bạn có thể gọi lệnh Delete từ Index hoặc Details view. Chức năng này hoạt động như chúng ta đã mô tả.

ICT library

Tiêu đề	Tác giả	Nhà xuất bản	Năm xuất bản		
ASP.NET Core for dummy	Trump D.	Washington	2020	Details	Delete
Pro ASP.NET Core	Putin V.	Moscow	2020	Details	Delete
ASP.NET Core Video course	Obama B.	Washington	2020	Details	Delete
Programming ASP.NET Core MVC	Clinton B.	Washington	2020	Details	Delete
ASP.NET Core Razor Pages	Yelstin B.	Moscow	2020	Details	Delete

Tổng số: 5

<https://tuhocict.com>

ASP.NET Core for dummy

Tiêu đề ASP.NET Core for dummy

Tác giả Trump D.

Nhà xuất bản Washington

Năm xuất bản 2020

File

Tóm tắt

[Back to List](#)

[Delete](#)

<https://tuhocict.com>

Mỗi khi xóa một bản ghi, dữ liệu sẽ lưu vào file và điều hướng trở về trang Index.

Cập nhật dữ liệu

Chức năng cập nhật dữ liệu được chạy từ Index hoặc Details view và theo cùng quy trình như Delete:

1. Hiển thị form cập nhật;
2. Nếu người dùng chọn lưu kết quả => lưu vào file dữ liệu và điều hướng về Index;
3. Nếu người dùng muốn hủy lưu => điều hướng về Index hoặc Details.

Bước 1. Xây dựng action

Bổ sung hai action sau vào BookController.

```
1. public IActionResult Edit(int id) {
2.     var b = _service.Get(id);
3.     if (b == null) return NotFound();
4.     else return View(b);
5. }
6.
7. [HttpPost]
8. public IActionResult Edit(Book book) {
9.     if (ModelState.IsValid) {
10.         _service.Update(book);
11.         _service.SaveChanges();
12.         return RedirectToAction("Index");
13.     }
14.     return View(book);
15. }
```

Để ý rằng hai phương thức Edit ở đây cũng tuân theo mô hình Get-Post-Redirect giống như Delete ở trên.

Bước 2. Xây dựng Edit view

```
1. @model BookMan.Mvc.Models.Book
2.
3. @{
4.     ViewData["Title"] = $"Editing {Model.Name}";
5. }
6.
7. <p class="h3">You're editing: <strong>@Model.Name</strong></p>
8. <hr />
9.
10. <form asp-action="Edit" enctype="multipart/form-data">
11.
12.     <div class="row">
13.         <input type="hidden" asp-for="Id" />
14.         <div class="col-md-6">
15.             <div class="form-group">
16.                 <label asp-for="Name" class="control-label"></label>
17.                 <input asp-for="Name" class="form-control" />
18.                 <span asp-validation-for="Name" class="text-danger small"></span>
19.             </div>
20.             <div class="form-group">
21.                 <label asp-for="Authors" class="control-label"></label>
22.                 <input asp-for="Authors" class="form-control" />
23.                 <span asp-validation-for="Authors" class="text-danger small"></span>
24.             </div>
25.             <div class="form-group">
26.                 <label asp-for="Publisher" class="control-label"></label>
27.                 <input asp-for="Publisher" class="form-control" />
28.                 <span asp-validation-for="Publisher" class="text-danger small"></span>
29.             </div>
30.             <div class="form-group">
31.                 <label asp-for="Year" class="control-label"></label>
32.                 <input asp-for="Year" class="form-control" />
33.                 <span asp-validation-for="Year" class="text-danger small"></span>
34.             </div>
35.         </div>
36.     </div>
```

```

36.         <div class="col-md-6">
37.             <div class="form-group">
38.                 <label asp-for="Description" class="control-label"></label>
39.                 <textarea asp-for="Description" class="form-control" rows="8"></textarea>
40.                 <span asp-validation-for="Description" class="text-danger small"></span>
41.             </div>
42.             <div class="form-group">
43.                 <input type="file" name="file" />
44.                 <input type="hidden" asp-for="DataFile" />
45.             </div>
46.             <div asp-validation-summary="ModelOnly" class="text-danger"></div>
47.         </div>
48.     </div>
49.
50.     @section Scripts {
51.         @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
52.     }
53.
54.     <hr />
55.     <div class="form-group">
56.         <input type="submit" value="Save" class="btn btn-sm btn-outline-primary" /> |
57.         <a asp-action="Details" asp-route-id="@Model.Id" class="btn btn-sm btn-outline-i
58.         <a asp-action="Index" class="btn btn-sm btn-outline-secondary">Back to List</a>
59.     </div>
60. </form>

```

Để ý một số thẻ span có tham số **asp-validation-for**. Đây là hỗ trợ thẩm định dữ liệu (input validation) trong ASP.NET Core. Cơ chế này giúp kiểm tra xem các property của model có phù hợp yêu cầu hay không.

Một số yêu cầu chúng ta đã chỉ định khi **xây dựng domain model** bao gồm [Required] và [Range] .

Chúng ta sẽ học chi tiết về thẩm định dữ liệu trong một bài học riêng.

Bước 3. Thêm nút lệnh Edit

Thêm nút lệnh Edit với thẻ a như sau vào Index view:

```

1. <a asp-action="Edit" asp-route-id="@item.Id" class="btn btn-sm btn-outline-warning">Edit

```

Thêm nút lệnh sau vào Details view:

```

1. <a asp-action="Edit" asp-route-id="@Model.Id" class="btn btn-sm btn-outline-warning">Edi

```

Giờ bạn có thể chạy thử ứng dụng với chức năng cập nhật dữ liệu.

ICT library

Tiêu đề	Tác giả	Nhà xuất bản	Năm xuất bản			
ASP.NET Core for dummy	Trump D.	Washington	2020	Details	Delete	Edit
Pro ASP.NET Core	Putin V.	Moscow	2020	Details	Delete	Edit
ASP.NET Core Video course	Obama B.	Washington	2020	Details	Delete	Edit
Programming ASP.NET Core MVC	Clinton B.	Washington	2020	Details	Delete	Edit
ASP.NET Core Razor Pages	Yelstin B.	Moscow	2020	Details	Delete	Edit

Tổng số: 5

<https://tuhocict.com>

You're editing: ASP.NET Core for dummy

Tiêu đề

ASP.NET Core for dummy

Tác giả

Trump D.

Nhà xuất bản

Washington

Năm xuất bản

2020

Tóm tắt

Chọn tệp

Không có tệp nào được chọn

Save

Cancel

Back to List

<https://tuhocict.com>

Chú ý, chức năng upload file chưa hoạt động. Chúng ta sẽ quay lại với chức năng này sau.

Thêm dữ liệu mới

Quy trình thêm dữ liệu diễn ra tương tự như Edit và Delete.

Bước 1. Thêm controller action

Thêm hai action sau đây vào BookController:

```
1. public IActionResult Create() => base.View(_service.Create());
2.
3. [HttpPost]
4. public IActionResult Create(Book book) {
5.     if (ModelState.IsValid) {
6.         _service.Add(book);
7.         _service.SaveChanges();
8.         return RedirectToAction("Index");
9.     }
10.    return View(book);
11. }
```

Bước 3. Xây dựng view

Tạo Create view với code như sau:

```
1. @model BookMan.Mvc.Models.Book
2.
3. @{
4.     ViewData["Title"] = "Create a new book";
5. }
6.
7. <p class="h3">Add a new book</p>
8. <hr />
9.
10. <form asp-action="Create" enctype="multipart/form-data">
11.
12.     <div class="row">
13.         <input type="hidden" asp-for="Id" />
14.         <div class="col-md-6">
15.             <div class="form-group">
16.                 <label asp-for="Name" class="control-label"></label>
17.                 <input asp-for="Name" class="form-control" />
18.                 <span asp-validation-for="Name" class="text-danger small"></span>
19.             </div>
```

```

20.         <div class="form-group">
21.             <label asp-for="Authors" class="control-label"></label>
22.             <input asp-for="Authors" class="form-control" />
23.             <span asp-validation-for="Authors" class="text-danger small"></span>
24.         </div>
25.         <div class="form-group">
26.             <label asp-for="Publisher" class="control-label"></label>
27.             <input asp-for="Publisher" class="form-control" />
28.             <span asp-validation-for="Publisher" class="text-danger small"></span>
29.         </div>
30.         <div class="form-group">
31.             <label asp-for="Year" class="control-label"></label>
32.             <input asp-for="Year" class="form-control" />
33.             <span asp-validation-for="Year" class="text-danger small"></span>
34.         </div>
35.     </div>
36.     <div class="col-md-6">
37.         <div class="form-group">
38.             <label asp-for="Description" class="control-label"></label>
39.             <textarea asp-for="Description" class="form-control" rows="8"></textarea>
40.             <span asp-validation-for="Description" class="text-danger small"></span>
41.         </div>
42.         <div class="form-group">
43.             <input type="file" name="file" />
44.             <input type="hidden" asp-for="DataFile" />
45.         </div>
46.         <div asp-validation-summary="ModelOnly" class="text-danger"></div>
47.     </div>
48. </div>
49.
50. @section Scripts {
51.     @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
52. }
53.
54. <hr />
55. <div class="form-group">
56.     <input type="submit" value="Create" class="btn btn-outline-primary btn-sm" /> |
57.     <a asp-action="Index" class="btn btn-sm btn-outline-secondary">Back to List</a>
58. </div>
59. </form>

```

Bước 3. Thêm nút lệnh Create

Thêm nút lệnh Create bằng khối html sau vào ngay trước bảng dữ liệu của Index view:

```

1. <div class="mb-3">
2.     <a asp-action="Create" class="btn btn-outline-primary">Create New</a>
3. </div>

```

Đến đây chức năng thêm dữ liệu đã hoàn thành. Bạn có thể chạy thử chương trình:

ICT library

Create New

Tiêu đề	Tác giả	Nhà xuất bản	Năm xuất bản			
ASP.NET Core for dummy	Trump D.	Washington	2020	Details	Delete	Edit
Pro ASP.NET Core	Putin V.	Moscow	2020	Details	Delete	Edit
ASP.NET Core Video course	Obama B.	Washington	2020	Details	Delete	Edit
Programming ASP.NET Core MVC	Clinton B.	Washington	2020	Details	Delete	Edit
ASP.NET Core Razor Pages	Yelstin B.	Moscow	2020	Details	Delete	Edit

Add a new book

Tiêu đề

Lập trình ASP.NET Core nâng cao

Tác giả

Mai Chi

Nhà xuất bản

ICT

Năm xuất bản

2020

Tóm tắt

Chọn tệp

Không có tệp nào được chọn

Create

Back to List

<https://tuhocict.com>

Lưu ý chức năng upload file chưa hoạt động. Chúng ta sẽ quay lại với chức năng này sau.

Tái sử dụng code với partial view

Khi thực hiện bài thực hành này bạn có thể để ý thấy rằng có vài chỗ đang bị lặp code. Cụ thể, Details và Delete view cùng chia sẻ một khối code razor hiển thị thông tin chi tiết của một cuốn sách. Create và Edit view cùng chia sẻ code của form nhập liệu.

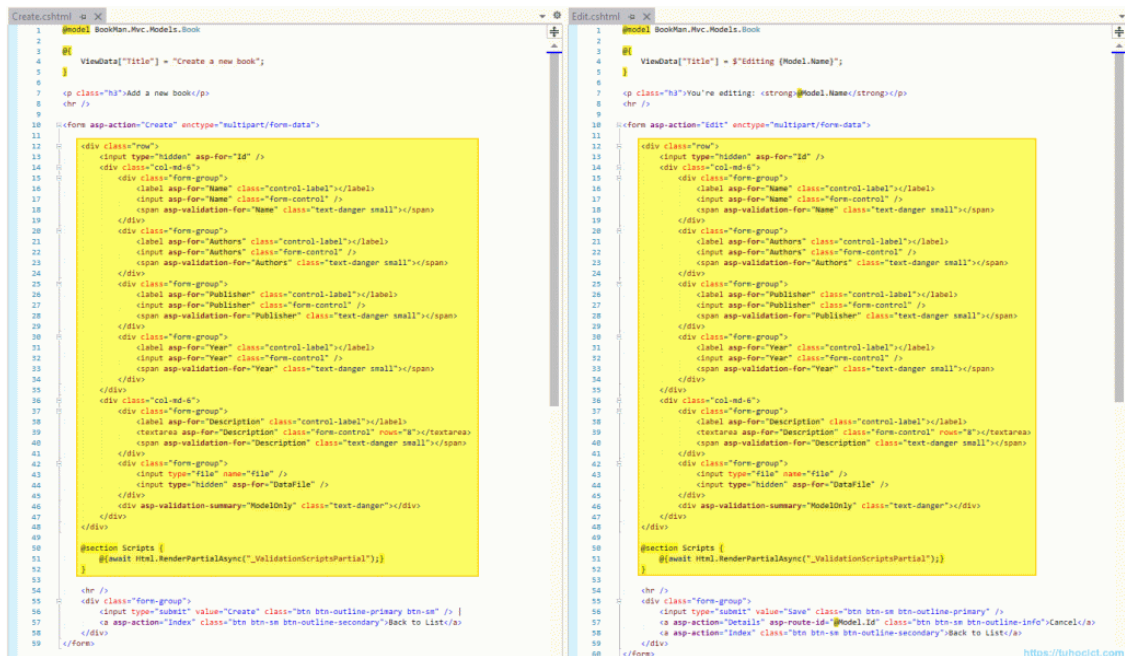
Delete.cshtml

1 @model BookMan.Mvc.Models.Book
2
3 @{
4 ViewData["Title"] = "Confirmation";
5 }
6
7 <p class="p-3 alert-danger">Are you sure you want to delete
8 this?</p>
9
10 <div>
11 <p class="h3">@Model.Name</p>
12 <hr />
13 <dl class="row">
14 <dt class="col-sm-2"><label asp-for="Name"></
15 label></dt>
16 <dd class="col-sm-10">@Model.Name</dd>
17 <dt class="col-sm-2"><label asp-for="Authors"></
18 label></dt>
19 <dd class="col-sm-10">@Model.Authors</dd>
20 <dt class="col-sm-2"><label asp-for="Publisher"></
21 label></dt>
22 <dd class="col-sm-10">@Model.Publisher</dd>
23 <dt class="col-sm-2"><label asp-for="Year"></
24 label></dt>
25 <dd class="col-sm-10">@Model.Year</dd>
26 <dt class="col-sm-2"><label asp-for="DataFile"></
27 label></dt>
28 <dd class="col-sm-10">@Model.DataFile</dd>
29 <dt class="col-sm-2"><label asp-
30 for="Description"></label></dt>
31 <dd class="col-sm-10">@Model.Description</dd>
32 </dl>
33 </div>
34 <hr />
35 <form asp-action="Delete">
36 <input type="hidden" asp-for="Id" />
37 <input type="submit" value="Delete" class="btn btn-
38 outline-danger btn-sm" /> |
39 <a asp-action="Index" class="btn btn-sm btn-outline-
40 secondary">Back to List
41 </form>

Details.cshtml

1 @model BookMan.Mvc.Models.Book
2
3 @{
4 ViewData["Title"] = Model.Name;
5 }
6
7 <div>
8 <p class="h3">@Model.Name</p>
9 <hr />
10 <dl class="row">
11 <dt class="col-sm-2"><label asp-for="Name"></
12 label></dt>
13 <dd class="col-sm-10">@Model.Name</dd>
14 <dt class="col-sm-2"><label asp-for="Authors"></
15 label></dt>
16 <dd class="col-sm-10">@Model.Authors</dd>
17 <dt class="col-sm-2"><label asp-for="Publisher"></
18 label></dt>
19 <dd class="col-sm-10">@Model.Publisher</dd>
20 <dt class="col-sm-2"><label asp-for="Year"></
21 label></dt>
22 <dd class="col-sm-10">@Model.Year</dd>
23 <dt class="col-sm-2"><label asp-for="DataFile"></
24 label></dt>
25 <dd class="col-sm-10">@Model.DataFile</dd>
26 <dt class="col-sm-2"><label asp-
27 for="Description"></label></dt>
28 <dd class="col-sm-10">@Model.Description</dd>
29 </dl>
30 </div>
31 <hr />
32 <div>
33 <a asp-action="Index" class="btn btn-sm btn-outline-
34 secondary">Back to List
35 <a asp-action="Delete" asp-route-id="@Model.Id"
36 class="btn btn-sm btn-outline-danger">Delete
37 <a asp-action="Edit" asp-route-id="@Model.Id"
38 class="btn btn-sm btn-outline-warning">Edit
39 </div>

Details và Delete cùng chia sẻ một khối code



Edit và Create có cùng khối code thân form

Giữa Delete và Details view cùng chia sẻ một logic hiển thị. Tương tự, Edit và Create view có cùng logic hiển thị. Do vậy, sự lặp code ở đây không phải là vấn đề lớn. Chúng ta hoàn toàn có thể chấp nhận vấn đề này. Đặc biệt, nếu bạn mong muốn sự tiện lợi của scaffolding, bạn có thể chấp nhận việc lặp code giữa hai cặp view trên.

Tuy nhiên, nếu bạn ưa thích sự tối ưu, việc lặp code như trên là khó chấp nhận. Trong tình huống này bạn có thể giải quyết sự lặp code bằng cách áp dụng partial view.

Hãy cùng thực hiện các bước sau đây để vận dụng partial view và xem cách chống lặp code với partial view.

Bước 1. Tạo partial view chung cho Details và Delete view

Tạo _Single.cshtml trong thư mục /Views/Book và copy đoạn code trùng nhau của Details và Delete view sang. Bạn thu được file _Single.cshtml như sau:

```
1. @model BookMan.Mvc.Models.Book
2. <div>
3.     <p class="h3">@Model.Name</p>
4.     <hr />
5.     <dl class="row">
6.         <dt class="col-sm-2"><label asp-for="Name"></label></dt>
7.         <dd class="col-sm-10">@Model.Name</dd>
8.         <dt class="col-sm-2"><label asp-for="Authors"></label></dt>
9.         <dd class="col-sm-10">@Model.Authors</dd>
10.        <dt class="col-sm-2"><label asp-for="Publisher"></label></dt>
11.        <dd class="col-sm-10">@Model.Publisher</dd>
12.        <dt class="col-sm-2"><label asp-for="Year"></label></dt>
13.        <dd class="col-sm-10">@Model.Year</dd>
14.        <dt class="col-sm-2"><label asp-for="DataFile"></label></dt>
15.        <dd class="col-sm-10">@Model.DataFile</dd>
16.        <dt class="col-sm-2"><label asp-for="Description"></label></dt>
17.        <dd class="col-sm-10">@Model.Description</dd>
18.    </dl>
19. </div>
```

Bước 2. Cập nhật Delete và Details view

Xóa bỏ phần code bôi vàng và thay bằng

```
1. <partial name="_Single" model="Model" />
```

Khi này code của Details.cshtml và Delete.cshtml còn như sau:

	Details.cshtml	Delete.cshtml
1.	@model BookMan.Mvc.Models.Book	
2.		
3.	{	
4.	ViewData["Title"] = Model.Name;	
5.	}	
6.		
7.	<partial name="_Single" model="Model" />	
8.		
9.	<hr />	
10.		
11.	<div>	
12.	<a asp-action="Index" class="btn btn-sm btn-outline-secondary">Back to List	
13.	<a asp-action="Delete" asp-route-id="@Model.Id" class="btn btn-sm btn-outline-danger">Delete	
14.	<a asp-action="Edit" asp-route-id="@Model.Id" class="btn btn-sm btn-outline-warning">Edit	
15.	</div>	

Bước 3. Cập nhật Edit và Create view

Tương tự, copy phần bôi vàng trùng nhau của Edit và Create view sang file mới _Form.cshtml trong /Views/Book. Bạn thu được file _Form.cshtml có code như sau:

```
1. @model BookMan.Mvc.Models.Book
2.
3. <div class="row">
4.     <input type="hidden" asp-for="Id" />
5.     <div class="col-md-6">
6.         <div class="form-group">
7.             <label asp-for="Name" class="control-label"></label>
8.             <input asp-for="Name" class="form-control" />
9.             <span asp-validation-for="Name" class="text-danger small"></span>
10.        </div>
11.        <div class="form-group">
12.            <label asp-for="Authors" class="control-label"></label>
13.            <input asp-for="Authors" class="form-control" />
14.            <span asp-validation-for="Authors" class="text-danger small"></span>
15.        </div>
16.        <div class="form-group">
17.            <label asp-for="Publisher" class="control-label"></label>
18.            <input asp-for="Publisher" class="form-control" />
19.            <span asp-validation-for="Publisher" class="text-danger small"></span>
20.        </div>
21.        <div class="form-group">
22.            <label asp-for="Year" class="control-label"></label>
23.            <input asp-for="Year" class="form-control" />
24.            <span asp-validation-for="Year" class="text-danger small"></span>
25.        </div>
26.    </div>
27.    <div class="col-md-6">
28.        <div class="form-group">
29.            <label asp-for="Description" class="control-label"></label>
30.            <textarea asp-for="Description" class="form-control" rows="8"></textarea>
31.            <span asp-validation-for="Description" class="text-danger small"></span>
32.        </div>
33.        <div class="form-group">
34.            <input type="file" name="file" />
35.            <input type="hidden" asp-for="DataFile" />
36.        </div>
37.        <div asp-validation-summary="ModelOnly" class="text-danger"></div>
38.    </div>
39. </div>
40.
41. @section Scripts {
42.     @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
43. }
```

Bước 4. Cập nhật Edit và Create view

Thay cho khối code đã di chuyển sang _Form.cshtml bằng lệnh sau:

```
1. <partial name="_Form" model="Model" />
```

Khi này, code của Edit và Create view trở thành như sau:

Khi bạn chạy thử ứng dụng, các chức năng không thay đổi.

Tuy nhiên, code của các view tương ứng đã được đơn giản hóa đáng kể và không còn lặp code nữa.

Ở đây bạn đã gặp cách sử dụng partial view để tránh lặp code Razor. Bạn sẽ học chi tiết hơn về partial view trong một bài học riêng.

Partial view là một tính năng của ASP.NET Core cho phép tái sử dụng một khối code Razor đặt trong một file riêng. Bạn có thể hình dung partial view không khác biệt gì so với một view thông thường trong MVC. Thay vì chỉ định trong action, bạn có thể chèn partial view vào một view khác qua thẻ `<partial />` như chúng ta đã làm ở trên.

Kết luận

Trong bài học này chúng ta đã thực hành xây dựng một ứng dụng đơn giản với các chức năng CRUD dữ liệu cơ bản.

Như bạn đã thấy, xây dựng các chức năng CRUD dữ liệu trong ASP.NET Core MVC tương đối đơn giản.

Riêng đối với nhóm Delete – Update – Create bạn cần sử dụng mô hình Get – Post – Redirect.

Một số chức năng cao cấp hơn chúng ta sẽ để dành cho bài thực hành thứ 3.

Link tải mã nguồn: https://1drv.ms/u/s!Ar_aj4rIJ2qGkf8rCeoYjbiNZRbg4w?e=gHgJOg

- + Nếu bạn thấy site hữu ích, trước khi rời đi hãy **giúp đỡ** site bằng một hành động nhỏ để site có thể phát triển và phục vụ bạn tốt hơn.
 - + Nếu bạn thấy bài viết hữu ích, hãy giúp **chia sẻ** tới mọi người.
 - + Nếu có thắc mắc hoặc cần trao đổi thêm, mời bạn viết trong phần **thảo luận** cuối trang.
- Cảm ơn bạn!