Lab 2: Tạo trang chỉnh sửa Sách (Book Editting)

♣ Muc tiêu:

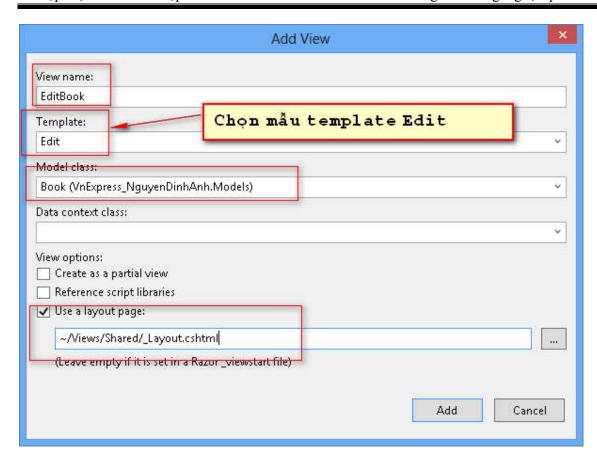
- Hiểu và nắm bắt được cơ chế xử lý của MVC
- Hiểu về View Model Controller
- ✓ Tạo mới một Method ActionResult có tên là EditBook

<u>Mô tả:</u> Hàm này nhận vào **ID** của Book sau đó tìm kiếm Book trong dữ liệu ListBook được khởi tạo ban đầu, nếu có cuốn sách đó thì trả Cuốn sách đó (**book**) tới **View EditBook** để chỉnh sửa, nếu không tìm thấy thì trả về trang **NotFound**

Code Mô tả của hàm

```
//Get Book/EditBook/Id
Oreferences
public ActionResult EditBook(int id)
{
    var books = new ListcBook>();
    books.Add(new Book(1, "HTML5 & CSS3 The complete Manual", "Author Name Book 1", "/Content/Images/booklcover.png"));
    books.Add(new Book(2, "HTML5 & CSS Responsive web Design cookbook", "Author Name Book 2", "/Content/Images/book2cover.png"));
    books.Add(new Book(3, "Professional ASP.NET MVC5", "Author Name Book 3", "/Content/Images/book3cover.png"));
    //Check if not exit
    Book book = new Book();
    foreach (Book b in books)
    {
        if(b.Id=id)
        {
            book = b;
            break;
        }
        if (book == null)
        {
            return HttpNotFound();
        }
        return View(book);
}
```

Tạo View **EditBook** như bình thường, bằng cách nhấn chuột phải vào **EditBook** và chọn **Add New View**

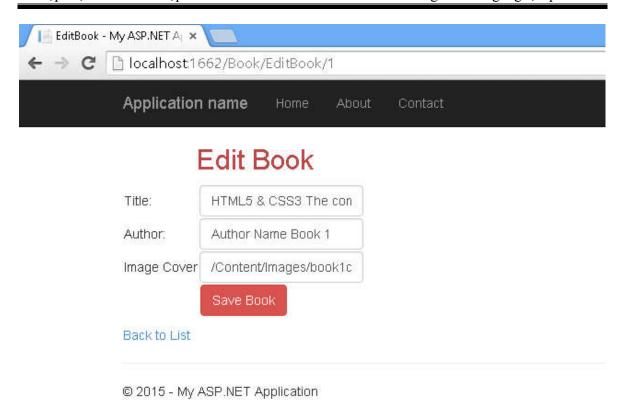


Chỉnh sửa trang **EditBook** theo như mẫu bên dưới để hiển thị trang sửa sách theo yêu cầu của người dùng

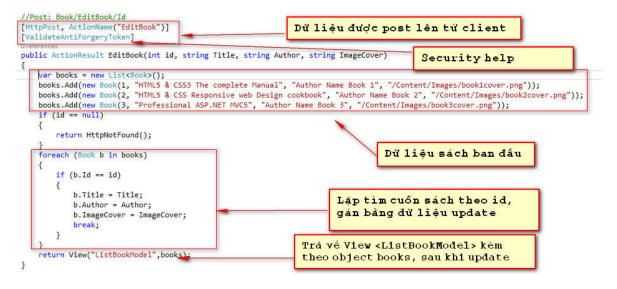
```
model VnExpress_NguyenDinhAnh.Models.Book
   ViewBag.Title = "EditBook";
                                                                 Security help
   Layout = "~/Views/Shared/_Layout.cshtml";
<h2 style="padding-left:80px" class="text-danger">Edit Book</ba>
@using (Html.BeginForm())
                                           Validation, kiểm tra sự ràng buộc dữ liệu của
   @Html.AntiForgeryToken()
                                           các Field Book
  @Html.ValidationSummary(true
   @Html.HiddenFor(model => model.Id)
   (tr)
                                         Biến ấn Iđ của Book sẽ dùng để trả về server lưu dữ
         Title:
                                         liệu
         @Html.TextBoxFor(model => model.Title, new { @class = "form-control", placeholder = "Title" })
      Author:
         @Html.TextBoxFor(model => model.Author, new { @class = "form-control", placeholder = "Author" })
      Image Cover
         @Html.TextBoxFor(model => model.ImageCover, new { @class = "form-control", placeholder = "Author" })

     <input type="submit" value="Save Book" class="btn btn-danger"</pre>
      <div>
   @Html.ActionLink("Back to List", "Index")
</div>
```

Giao diện trang EditBook sau khi đã chỉnh sửa Layout



Viết hàm xử lý lưu dữ liệu sách từ người dùng (chú ý: bổ sung phương thức **Set** cho **Model Book** ở trên)



- **♣** Tạo trang nhập sách, có sử dụng cơ chế ràng buộc kiểm tra dữ liệu (Validation) trong ứng dụng Web MVC
- ✓ Muc tiêu:
 - Biết cách sử dụng Model để gửi dữ liệu từ Client về Server
 - Hiểu rõ và biết cách sử dụng các kiểm tra ràng buộc

Trong **BookController** bổ sung thêm **Method Create** để mở **View thêm** mới sách và Nhận dữ liệu từ người dùng trả về

```
Oreferences

public ActionResult CreateBook()

{
    return View();
}
```

Tạo View CreateBook có kiểm tra ràng buộc về dữ liệu người dùng nhập vào

```
@model VnExpress NguyenDinhAnh.Models.Book
   ViewBag.Title = "CreateBook";
    Layout = "~/Views/Shared/_Layout.cshtml"
   <script src="~/Scripts/jquery-1.10.2.min.js"></script>
    <script src="~/Scripts/jquery.validate.min.js"></script>
    <script src="~/Scripts/jquery.validate.unobtrusive.js"></script>
<h2>Create Book</h2>
@using (Html.BeginForm())
                                                  Bổ sung thư viện javascript kiểm tra
                                                  validation phía client
   @Html.AntiForgeryToken()
   <div class="form-horizontal">
        <h4>Book</h4>
                                                                             Kiểm tra ràng buộc
        <hr />
                                                                             dữ liệu client
       @Html.ValidationSummary(true)
        <div class="form-group">
           @Html.LabelFor(model => model.Title, new { @class = "contrg/1-label col-md-2" })
           <div class="col-md-10">
                html.EditorFor(model => model.Title)
               @Html.ValidationMessageFor(model => model.Title)
           </div
        </div>
        <div class="form-group">
           @Html.LabelFor(model => model.Author, new { @class = "control-label col-md-2" })
           <div class="col-md-10">
               @Html.EditorFor(model => model.Author)
               @Html.ValidationMessageFor(model => model.Author)
        </div>
        <div class="form-group">
           @Html.LabelFor(model => model.ImageCover, new { @class = "control-label col-md-2" })
           <div class="col-md-10">
               @Html.EditorFor(model => model.ImageCover)
               @Html.ValidationMessageFor(model => model.ImageCover)
           </d vs
        </div>
        <div class="form-group">
           <div class="col-md-offset-2 col-md-10">
               <input type="submit" value="Create" class="btn btn-default" />
           </div>
       </div>
   </div>
}
   @Html.ActionLink("Back to List", "Index")
</div>
```

Để quy định ràng buộc về dữ liệu từ người dùng nhập vào, ta sử dụng lớp System.ComponentModel.DataAnnotations để hỗ trợ.

Sửa Model Book, để thêm một số ràng buộc dữ liệu cho người dùng nhập vào

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;
namespace VnExpress_NguyenDinhAnh.Models
   18 references
   public class Book
       private int id;
       private string title;
       private string author;
       private string image_cover;
       public Book()
       9 references
       public Book(int id, string title, string author, string image_cover)
            this.id = id:
            this.title = title;
           this.author = author;
           this.image_cover = image cover;
                                     Kiểm tra ràng buộc cho tiêu đề
       }

    Không được trống

        -references
       public int Id

    Tối đa là 250 ký tự

            get { return id; }
        [Required(ErrorMessage = "Tiêu đề không được trống")]
        [StringLength(250, ErrorMessage = "Tiêu đề sách không được vượt quá 250 ký tự")]
        [Display(Name = "Tiêu đề")]
       public string Title
            get { return title; }
           set { title = value; }
       public string Author
            get { return author; }
           set { author = value; }
       public string ImageCover
            get { return image_cover; }
           set { image cover = value; }
   }
1
```

Viết hàm **CreateBook** (**HttpPost**), để nhận giá trị trả về từ **Client** và lưu dữ liệu, <u>chú ý:</u> bổ sung thêm phương thức **Set** cho thuộc tính **Id của Model Book**

Màn hình giao diện nhập sách sau khi đã hoàn chỉnh

Create Book

Mã sách	D không được trống
Tiêu đề	Tiêu đề không được trống
Tác giả	
Đường dẫn Ảnh Bìa	
	Create
ck to List	

Màn hình giao diện nhập sách

Create Book

© 2015 - Nguyen Dinh Anh

Mã sách	5
Tiêu đề	Lap MVC
Tác giả	Nguyen Dinh Anh
Đường dẫn Ánh Bìa	/Content/images/cover5
	Create
Back to List	

Biên soạn: ThS. Nguyễn Đình Ánh