Lab 4: Tạo trang nhập sách, có sử dụng cơ chế ràng buộc kiểm tra dữ liệu (Validation) trong ứng dụng Web MVC

♣ Mục 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 />
                                                                              đữ liệu client
        @Html.ValidationSummary(true)
        <div class="form-group">
           @Html.LabelFor(model => model.Title, new { @class = "contrg1-labe1 col-md-2" })
            <div class="col-md-10">
                <mark>@</mark>Html.EditorFor(model => model.Title)
               MHtml.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">
                MHtml.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)
               MHtml.ValidationMessageFor(model => model.ImageCover)
            </d tys
        </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>
}
<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 liêu 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
        }

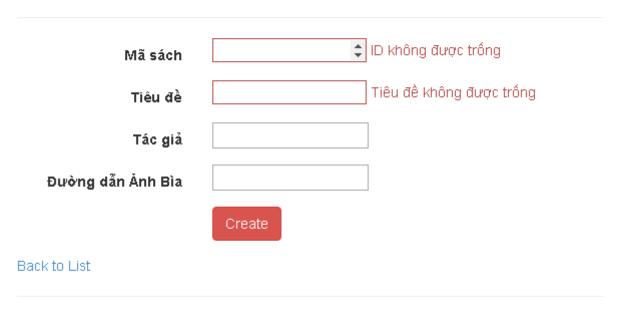
    Không được trống

        -references
        public int Id
                                     - Tố1 đã 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")]
        1 reference
        public string Title
            get { return title; }
            set { title = value; }
        public string Author
        {
            get { return author; }
            set { author = value; }
        1 reference
        public string ImageCover
            get { return image_cover; }
            set { image cover = value; }
    }
}
```

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



© 2015 - Nguyen Dinh Anh

Màn hình giao diện nhập sách

Create Book

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

© 2015 - Nguyen Dinh Anh