

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

```

References
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())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Book</h4>
        <hr />
        @Html.ValidationSummary(true)

        <div class="form-group">
            @Html.LabelFor(model => model.Title, new { @class = "control-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>

        <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)
            </div>
        </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>
```

Bổ sung thư viện javascript kiểm tra validation phía client

Kiểm tra ràng buộc dữ liệu client

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;
        1 reference
        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;
        }
        -references
        public int Id
        {
            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; }
        }
        1 reference
        public string Author
        {
            get { return author; }
            set { author = value; }
        }
        1 reference
        public string ImageCover
        {
            get { return image_cover; }
            set { image_cover = value; }
        }
    }
}

```

Kiểm tra ràng buộc cho tiêu đề
 - Không được trống
 - Tối đa là 250 ký tự

[Required(ErrorMessage = "Tiêu đề không được trống")]
 [StringLength(250, ErrorMessage = "Tiêu đề sách không được vượt quá 250 ký tự")]
 [Display(Name = "Tiêu đề")]

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

```
[HttpPost, ActionName("CreateBook")]
[ValidateAntiForgeryToken]
public ActionResult Contact([Bind(Include = "Id ,Title, Author, ImageCover")]Book book)
{
    var books = new List<Book>();
    //Sách mặc định
    books.Add(new Book(1, "HTML5 & CSS3 The complete Manual", "Author Name Book 1", "/Content/Images/book1cover.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"));
    try
    {
        if (ModelState.IsValid)
        {
            //Thêm mới sách
            books.Add(book);
        }
    }
    catch (RetryLimitExceededException /* dex */)
    {
        ModelState.AddModelError("", "Error Save Data");
    }
    //Trả về trang xem sách với danh sách Book mới
    return View("ListBookModel", books);
}
```

Màn hình giao diện nhập sách sau khi đã hoàn chỉnh

Create Book

Mã sách

ID không được trống

Tiêu đề

Tiêu đề không được trống

Tác giả

Đường dẫn Ảnh Bìa

Create

[Back to List](#)

© 2015 - Nguyen Dinh Anh

Màn hình giao diện nhập sách

Create Book

Mã sách	<input type="text" value="5"/>
Tiêu đề	<input type="text" value="Lap MVC"/>
Tác giả	<input type="text" value="Nguyen Dinh Anh"/>
Đường dẫn Ảnh Bìa	<input type="text" value="/Content/images/cover5"/>

Create

[Back to List](#)

© 2015 - Nguyen Dinh Anh