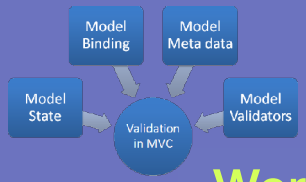


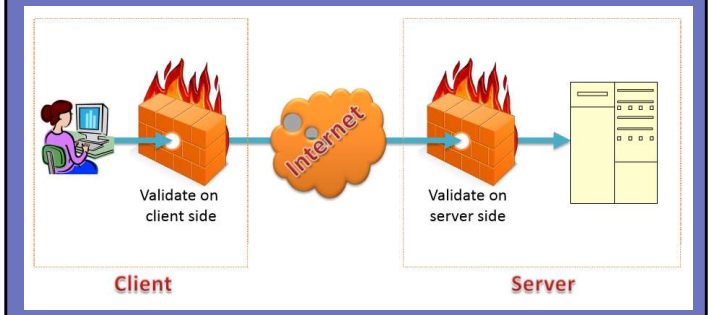
ASP.NET MVC Working with Data

Data Validation

Table of Contents

- Data Validation
  - Annotation attributes
- Session, TempData and Cache
- Application, Server, Cookie

Validation model



Validation with Annotations

- Attributes defined in System.ComponentModel.DataAnnotations
- Covers common validation patterns
  - Required
  - StringLength
  - Regex
  - Range

```
public class LogOnModel
{
    [Required]
    public string UserName { get; set; }

    [Required]
    public string Password { get; set; }

    public bool RememberMe { get; set; }
}
```

5

Data Validation Attributes

Attribute	Description
Compare	Checks whether two specified properties in the model have the same value.
CustomValidation	Checks the value against the specified custom function.
EnumDataType	Checks whether the value can be matched to any of the values in the specified enumerated type.
Range	Checks whether the value falls in the specified range. It defaults to numbers, but it can be configured to consider a range of dates, too.
RegularExpression	Checks whether the value matches the specified expression.
Remote	Makes an Ajax call to the server, and checks whether the value is acceptable.
Required	Checks whether a non-null value is assigned to the property. It can be configured to fail if an empty string is assigned.
StringLength	Checks whether the string is longer than the specified value.

6

Validating Model – Controller (server)

- ModelState.IsValid – will give us information about the data validation success
- ModelState.AddModelError – custom error

```
[HttpPost]
public ActionResult Edit(ForumPosts forumPost)
{
    if (ModelState.IsValid)
    {
        if (forumPost.Author != "Nikolay.IT")
        {
            ModelState.AddModelError("Author", "Wroooooong!");
        }
        db.Entry(forumPost).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(forumPost);
}
```

7

Validating Model – View (client)

- @Html.ValidationSummary – output errors
- @Html.ValidationMessageFor(…) – outputs validation message for specified property

```
@using (Html.BeginForm()) {
    @Html.ValidationSummary(true)

    <div class="editor-label">
        @Html.LabelFor(model => model.Title)
    </div>
    <div class="editor-field">
        @Html.EditorFor(model => model.Title)
        @Html.ValidationMessageFor(model => model.Title)
    </div>
}

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

Text box with integrated client-side validation

jQuery validation library required for unobtrusive JavaScript validation

8

Custom Validation

- Custom attributes
- Inherit ValidationAttribute

```
[AttributeUsage(AttributeTargets.Property)]
public sealed class MinLengthAttribute : ValidationAttribute
{

    // …

    public override bool IsValid(object value)
    {
        string valueAsString = value as string;
        return (valueAsString != null &&
                valueAsString.Length >= _minCharacters);
    }
}
```

9

Class-Level Model Validation

- Your model should implemented IValidatableObject
- From now on, MVC (works with EF too) will validate the object by your custom rules

```
public class Product : IValidatableObject
{
    public int      ProductID    { get; set; }
    public int      CategoryID   { get; set; }
    public string   ProductName  { get; set; }
    public Decimal? UnitPrice    { get; set; }
    public Int16?   UnitsInStock { get; set; }
    public Int16?   UnitsOnOrder { get; set; }
    public bool     Discontinued { get; set; }
    public virtual  Category Category { get; set; }

    //
    // Validate method that enforces two separate multi-property business rules

    public IEnumerable<ValidationResult> Validate(ValidationContext validationContext)
    {
        if ((UnitsOnOrder > 0) && (Discontinued))
            yield return new ValidationResult("Can't order discontinued products!", new [] { "UnitsOnOrder" });

        if ((UnitsInStock > 100) && (UnitsOnOrder > 0))
            yield return new ValidationResult("We already have a lot of these!", new [] { "UnitsOnOrder" });
    }
}
```

11

Custom Validation

- Example:

```
public sealed class EvenNumberAttribute: ValidationAttribute
{
    public EvenNumberAttribute() : base("Vui lòng nhập số chẵn !") { }

    public override bool IsValid(object value)
    {
        if (value == null)
        {
            return true;
        }
        return Convert.ToInt64(value) % 2 == 0;
    }
}
```

```
[EvenNumber]
public String Age{get;set}
```

Other Annotations

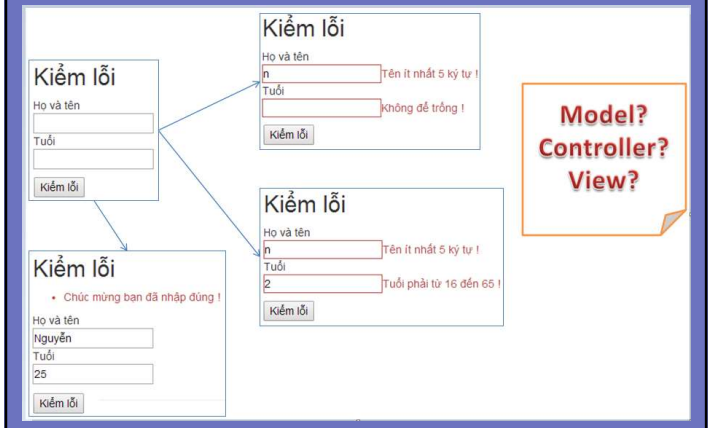
```
[Bind(Exclude = "Id")]
public class Album
{
    public Guid Id { get; set; }

    [Required(ErrorMessage = "An Album Title is required")]
    [StringLength(160)]
    public string Title { get; set; }

    public Genre Genre { get; set; }
    public Artist Artist { get; set; }

    [Required(ErrorMessage = "Price is required")]
    [Range(0.01, 100.00, ErrorMessage = "Price must be between 0.01 and 100.00")]
    public Decimal Price { get; set; }
    public string AlbumArt { get; set; }

    public Album() { }

    public Album(DynamicContent AlbumItem)
    {
        // save simple values
        this.Id = AlbumItem.Id;
        this.Title = AlbumItem.GetValue<string>("Title");
        this.Price = AlbumItem.GetValue<decimal>("Price");

        var mgr = DynamicModuleManager.GetManager();
```

Display / Edit Annotations

Attribute	Description
DisplayColumn	Specify the property of a model class for simple text display.
HiddenInput	Render value in a hidden input (when editing).
UIHint	Specify the name of the template to use for rendering.
DataType	Common templates (email, password, URL, currency)
ReadOnly	Specify a read-only property (for model binding).
DisplayFormat	Format strings and null display text
ScaffoldColumn	Turn off display and edit capabilities
DisplayName	Friendly name for labels
Bind	Tells the model binder which properties to include/exclude

13

Example



15

Annotation example

Annotation	Ví dụ
[Required]	[Required] public String Name{get;set}
[Range(Min, Max)]	[Range(16, 65)] public String Age{get;set}
[StringLength(Max)]	[StringLength (20, MinimumLength=5)] public String Password{get;set}
[EmailAddress]	[EmailAddress] public String Email{get;set}
[CreditCard]	[CreditCard] public String CardNumber{get;set}
[Url]	[Url] public String Website{get;set}
[Compare(Property)]	[Compare("Password")] public String ConfirmPassword{get;set}
[RegularExpression(Regex)]	[RegularExpression("\d{9}")] public String IdCard{get;set}
[MinLength(Min)]	[MinLength(1)] public String[] Hobbies{get;set}
[MaxLength (Max)]	[MaxLength (255)] public String Description{get;set}

14

Example

- Model:

```
public class EmployeeInfo
{
    [MinLength(5, ErrorMessage="Tên ít nhất 5 ký tự !")]
    public String FullName { get; set; }
    [Required(ErrorMessage="Không để trống !")]
    [Range(16, 65, ErrorMessage = "Tuổi phải từ 16 đến 65 !")]
    public int Age { get; set; }
}
```

16

Example

- Controller:

```
public class ValidatorController : Controller
{
    public ActionResult Index()
    {
        return View();
    }

    public ActionResult Validate(EmployeeInfo model)
    {
        if (ModelState.IsValid)
        {
            ModelState.AddModelError("", "Chúc mừng bạn đã nhập đúng !");
        }
        return View("Index");
    }
}
```

Kiểm lỗi phía server

Bổ sung thông báo lỗi model

Example

- Manual validation data:

```
public ActionResult Validate(String FullName, int Age)
{
    if (String.IsNullOrEmpty(FullName))
    {
        ModelState.AddModelError("FullName", "Không để trống họ và tên");
    }
    else if (FullName.Length < 5)
    {
        ModelState.AddModelError("FullName", "Ít nhất 5 ký tự !");
    }

    if (Age < 16 && Age > 65)
    {
        ModelState.AddModelError("Age", "Tuổi phải từ 16 đến 65 !");
    }

    if (ModelState.Count == 0) // không có lỗi nào
    {
        ModelState.AddModelError("", "Chúc mừng bạn đã nhập đúng !");
    }
    return View("Index");
}
```

Example

- View:

```
@model Mvc5CodeDemo.Models.EmployeeInfo
<h2>Kiểm lỗi</h2>
@Html.ValidationSummary(true)
@using (Html.BeginForm("Validate", "Validator"))
{
    <div>Họ và tên</div>
    @Html.TextBoxFor(m => m.FullName)
    @Html.ValidationMessageFor(m => m.FullName)
    <div>Tuổi</div>
    @Html.TextBoxFor(m => m.Age)
    @Html.ValidationMessageFor(m => m.Age)
    <hr />
    <input type="submit" value="Kiểm lỗi" />
}

@section scripts{
    @Scripts.Render("~/bundles/jqueryval")
}
```

Thông báo lỗi chung không bao gồm lỗi đã thông báo cho từng thuộc tính

Thông báo lỗi cho từng thuộc tính

Thực hiện kiểm lỗi phía client

Note

- Add @Html.AntiForgeryToken() to avoid forgery request

```
@using (Html.BeginForm("Withdraw", "Bank")) {
    @Html.AntiForgeryToken()
    <fieldset>
        <legend>Fields</legend>
        <p>
            <label for="Amount">Amount:</label>
            @Html.TextBox("Amount")
        </p>
        <p>
            <input type="submit" value="Withdraw" />
        </p>
    </fieldset>
}
```

Session, TempData and Cache

Session

- Use in:
  - Maintaining cart.
  - Maintain account login.
  - …
- Code:

Code in	Code	Example
Controller	Session	Session["A"]="Hello"
View	@Session	@Session["A"]
Class bất kỳ	HttpContext.Current.Session	HttpContext.Current.Session["A"]="Hello"

23

Session

- Each client has session id, which ASP.NET stores
- You can use it to store information in the memory of the application

```
var now = DateTime.Now;
if (this.HttpContext.Session["date"] == null)
{
    this.HttpContext.Session["date"] = now;
}

var date = (DateTime)this.HttpContext.Session["date"];
```

22

Session

Method	Example
Add(Key, Value)	Session.Add("Now", DateTime.Now)
[Key]=Value	Session["Cart"] = new ShoppingCart()
Remove(Key)	Session.Remove("Cart")
Clear()	Session.Clear()
Abandon()	Session.Abandon()
SessionID	Var id = Session.SessionID

24

TempData

- TempData can be used like a dictionary
- Each saved value lasts for the current and the next request
- Perfect for redirects

```
public ActionResult SaveToTempData()
{
    this.TempData["message"] = "Success!";
    return RedirectToAction("Redirection");
}

public ActionResult Redirection()
{
    var data = this.TempData["message"];
    return View(data);
}
```

25

Server, Application, Cookie

Cache

- You can save global data into the Cache
- It works like dictionary
- It is not per client, but rather global

```
public ActionResult SaveToCache()
{
    this.HttpContext.Cache["message"] = "Cache success!";
    return RedirectToAction("RedirectForCache");
}

public ActionResult RedirectForCache()
{
    var data = this.HttpContext.Cache["message"];
    return View(data);
}
```

26

Server

- `HttpServerUtility` – helper methods for processing HTTP requests (`Server` object)
  - `HtmlEncode(…)` – escapes given HTML, e.g. `"<img>"` → `"<img>"`
  - `HtmlDecode(…)` – un-escapes escaped HTML
  - `UrlEncode(…)` – encode string for the browser URL, e.g. `"+.net 4"` → `"%2B.net+4"`
  - `UrlDecode(…)` – decode url-encoded string
  - `MapPath(…)` – returns the server-side path for given resource given as relative path

28

Application

- Use in:
  - Count the number of visitors
  - Queue mails
  - Queue chat messages

29

Application

- Actions
  - Application. Add (name, value)
  - Application [name] = <value>
  - Application. Remove (name)
  - Application. Clear ()
  - Application. Lock ()
  - Application. Unlock ()

31

Application

- Retrieve Application Object:
  - In Controller: HttpContext.Application
  - In View: @ HttpContext.Current.Application
  - In any class: HttpContext.Current.Application

30

Application

- Example: (Global.asax)

```
    void Application_Start(object sender, EventArgs e)
    {
        // Code that runs on application startup
        Application.Lock();

        //Kiểm tra xem có file Dem.txt
        //Nếu chưa có, tạo file
        if (!System.IO.File.Exists(Server.MapPath("~/Dem.txt")))
            System.IO.File.WriteAllText(Server.MapPath("~/Dem.txt"), "0");

        //Nếu đã có file Dem.txt, đọc số liệu người truy cập
        Application["SoLuotTruyCap"] =
int.Parse(System.IO.File.ReadAllText(Server.MapPath("~/Dem.txt")));

        Application.UnLock();

    }
```

32

Application

- Example: (Global.asax)

```
void Session_Start(object sender, EventArgs e)
    {
        // Code that runs when a new session is started
        Application.Lock();
        //Tăng số lượt truy cập lên 1
        Application["SoLuotTruyCap"] = int.Parse(Application["SoLuotTruyCap"].ToString())
+ 1;
        Application.UnLock();
        //Ghi xuống file
        System.IO.File.WriteAllText(Server.MapPath("~/Dem.txt"),
Application["SoLuotTruyCap"].ToString());

        //Xử lý số người online
        //Nếu chưa có thì gán là 1, có rồi thì tăng 1
        Application.Lock();
        if (Application["SLOnline"] == null)//chưa có
            Application["SLOnline"] = 1;
        else
            Application["SLOnline"] = int.Parse(Application["SLOnline"].ToString()) + 1;
        Application.UnLock();
    }
```

33

Application

- Example: (Global.asax)

```
void Session_End(object sender, EventArgs e)
{
    // Code that runs when a session ends.
    // Note: The Session_End event is raised only when the sessionstate mode
    // is set to InProc in the Web.config file. If session mode is set to StateServer
    // or SQLServer, the event is not raised.
    Application.Lock();
    Application["SLOnline"] = int.Parse(Application["SLOnline"].ToString()) - 1;
    Application.UnLock();
}
```

34

Cookie

- Cookies are small text templates are stored on the client
- Get the cookie from client
  - In controller: Request.Cookies [name]
  - In View: @ Request.Cookies [name]

35

Cookie

- Send cookies on client
  - Response.Cookies.Add (cookie)
- create a cookie
  - HttpCookie cookie = new HttpCookie (name, value)
    - Create a cookie with the name and value
  - HttpCookie cookie = new HttpCookie (name)
    - Create a cookie with the name

36


Questions?