

Nội dung môn học

1. Giới thiệu
2. Các Control cơ bản trong ASPX
3. Ứng dụng ASP.NET
4. Quản lý trạng thái
5. Thao tác trên dữ liệu
6. Ajax
7. Webservice
8. Ứng dụng

Liên hệ

- Email: lucaotien@gmail.com
- Trao đổi: www.gocay.vn
- Tiêu đề mail đúng quy định:
[Lớp]-[MSSV]-[Tiêu đề]
- Tên file nộp bài tập cá nhân:
[Lớp]_[MSSV]_[BaiTapXXX].rar
- Tên file nộp bài tập nhóm:
[Lớp]_[NhomXX]_[MSSV1]_[MSSV2]_[...]
(***Tên file không dấu***)

Yêu cầu môn học

1. Tham gia **đầy đủ** các bài kiểm tra
 - LTWeb: Ktra trắc nghiệm
 - TH LT Web: Ktra trên máy+bài tập tuần
 - Đồ Án LTWeb: Tiến độ làm Đồ án
2. Nhóm 8 người, ngồi theo nhóm
3. Không làm việc riêng trong giờ học
4. Làm bài tập đầy đủ, nộp bài tại lớp (thu bài từ máy server vào đầu buổi học)

Tài liệu tham khảo

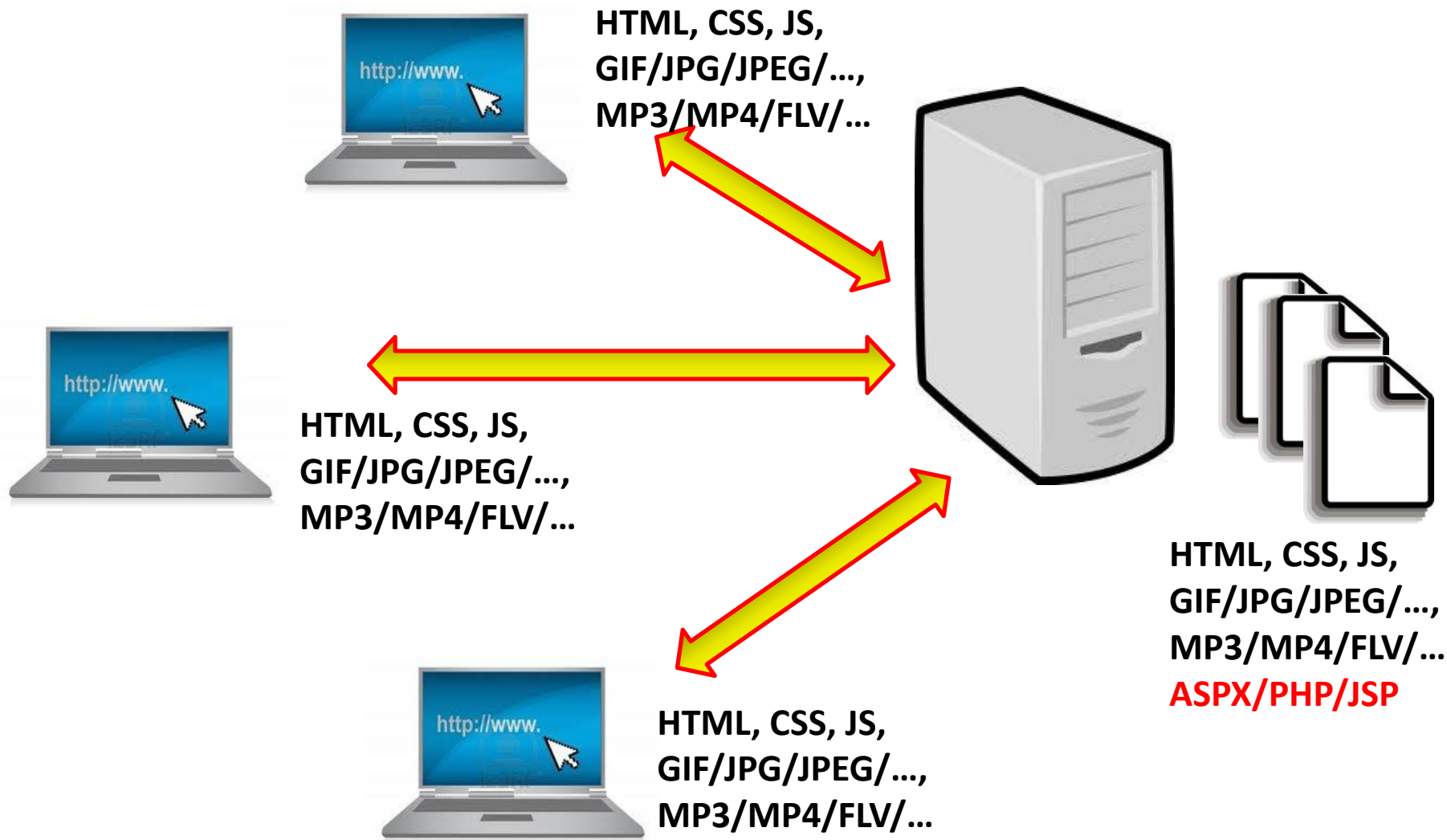
1. *Pro ASP.NET 4 in C# 2010 (fourth edition)*, Matthew MacDonald, Adam Freeman, and Mario Szpuszta.
2. *Beginning ASP.NET MVC 4*, José Rolando Guay Paz.



Tuần 01

Giới thiệu mô hình Client-WebServer & trang Web ASPX

Mô hình Client-WebServer



Xây dựng ASPX đầu tiên

- Chuẩn bị: cài đặt Visual Studio
- Thao tác:
 - Tạo project mới
 - Chạy ứng dụng (nhắm vững nguyên lý chạy)
 - Tạo control label và button.
 - Chạy ứng dụng.
 - Xử lý sự kiện cho button.
 - Chạy ứng dụng

Câu hỏi Tuần 1???

1. ASP.NET vs ASP?
2. Trang Web ASPX được thực thi như thế nào?
3. Control Label được biên dịch thành thẻ gì trong HTML?
4. Có thể LT một trang Web chỉ dung Label?

Nội dung thực hành tuần 01

- SiteMaster?
- Sử dụng layout HTML trên file .ASPX
 - Tạo layout trên DreamWeaver
 - Nhúng mã HTML vào trang ASPX
 - Cách sử dụng template có sẵn
- Bài tập:
 - Tạo layout theo mẫu 1(nhóm lẻ)
 - Tạo layout theo mẫu 2(nhóm chẵn)

Nội dung Đồ Án Tuần 01

- Chọn nhóm 2 người
- Chọn đề tài: ~~(có thể tự đề nghị)~~
 - **Website bán hàng**
 - ~~Website tin tức~~
 - ~~Website multimedia~~
 - ~~Website chia sẻ trực tuyến~~
 - ~~Blog cá nhân~~
 - ~~Diễn đàn học tập~~
- Đồ án cộng điểm:
 - Tìm hiểu mô hình 3 lớp và LINQ, ứng dụng **nhỏ** minh họa
 - Tìm hiểu mô hình MVC, ứng dụng **nhỏ** minh họa.
- Lớp trưởng lập danh sách nhóm
 - Thông tin nhóm: Họ tên, email, số đt liên lạc
 - Tên đề tài nhóm
 - Đồ án cộng điểm(nếu có)



Tuần 02

HTML form

Control trong ASPX

Ví dụ 2.1:

- Tạo trang .aspx blank
- Xử lý sự kiện Page_Load như sau:

```
protected void Page_Load(object sender,  
EventArgs e)  
{  
    string message = "<span style=\"color:Red\">"  
        + "Welcome To Me</span>";  
    Response.Write(message);  
}
```

Control trong ASPX

Ví dụ 2.2:

- Tạo trang .aspx với lblThongBao
- Xử lý sự kiện Page_Load như sau:

```
protected void Page_Load(object sender, EventArgs e)
{
    lblThongBao.Text = "Welcome To Me";
    lblThongBao.ForeColor = Color.Red;
}
```

(*Cần using System.Drawing*)

→Nhận xét???

Request-Response

Ví dụ 2.3:

Thao tác:

- Tạo trang ASPX (không dùng MasterPage)
- Sử dụng html để tạo trang web theo mẫu(Slide sau)
- Chạy xem kết quả

Trang ASPX:

```
<%@ Page Title="Home Page" Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head id="Head1" runat="server">
    <title>.:Welcome:.:</title>
</head>
<body>
    <form method="post" action="Default.aspx">
        <div>
            Nhập Họ Tên:
            <input type="text" name="txtHoTen" />
            <br />
            <input type="submit" value="Submit" id="OK" />
        </div>
    </form>
</body>
</html>
```

Trang ASPX.CS:

```
protected void Page_Load(object sender, EventArgs e)
{
    if (Request.Form["txtHoTen"] != null)
    {
        string sHoTen = Request.Form["txtHoTen"].ToString();
        Response.Write(sHoTen);
    }
}
```


Sự kiện

Ví dụ 2.4:

Thao tác:

- Tạo trang ASPX (không dùng MasterPage)
- Sử dụng control TextBox, Button để tạo trang web theo mẫu(Slide sau)
- Chạy xem kết quả, so sánh
 - Giống nhau?
 - Tính bảo mật?

Trang aspx:

```
<%@ Page Title="Home Page" Language="C#"
AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head id="Head1" runat="server">
    <title>:::Welcome:::</title>
</head>
<body>
    <form runat="server">
        <div>
            Nhập Họ Tên:
            <asp:TextBox ID="txtHoTen" runat="server">
        </asp:TextBox></br>
            <asp:Button ID="btnGui" runat="server" Text="Gửi"
            onclick="btnGui_Click" />
        </div>
    </form>
</body>
</html>
```

Trang.aspx.cs:

```
protected void btnGui_Click(object sender, EventArgs e)
{
    //string sHoTen = Request.Form["txtHoTen"].ToString();
    string sHoTen = txtHoTen.Text;
    Response.Write(sHoTen);
}
```

IsPostBack

- Thuộc tính của webpage có giá trị khi Web Form được load lại.
- Ví dụ 2.5
- Trang ASPX.CS:

```
protected void Page_Load(object sender, EventArgs e)
{
    if (IsPostBack != true)
    {
        string sHoTen = "Lu Cao Tien";
        Response.Write(sHoTen);
    }
}
```

Trang ASPX:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head id="Head1" runat="server">
    <title>:::Welcome::~.</title>
</head>
<body>
    <form runat="server">
        <div>

            Nhập Họ Tên:
            <asp:TextBox ID="txtHoTen" runat="server">
</asp:TextBox><br>
            <asp:Button ID="btnGui" runat="server" Text="Gửi"/>

        </div>
    </form>
</body>
</html>
```

Ví dụ 2.6:

Trang ASPX:

```
<%@ Page Title="Home Page" Language="C#"
AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head id="Head1" runat="server">
    <title>:::Welcome::~.</title>
</head>
<body>
    <form method="post" action="Default.aspx">
        <div>
            Nhập Họ Tên:
            <input type="text" name="txtHoTen" />
            <br />
            <input type="submit" value="Submit" id="OK" />
        </div>
    </form>
</body>
</html>
```

```
protected void Page_Load(object sender, EventArgs e)
{
    if (IsPostBack != null)
    {
        string sHoTen = "Lu Cao Tien";
        Response.Write(sHoTen);
    }
}
```

So sánh kết quả rút ra nhận xét.

Tổng kết

- HTML form và ASP form
 - Xử lý, sự kiện
 - Tính bảo mật
 - Thuộc tính IsPostBack
- Cho phép một số control HTML chạy như asp control
 - Run at server



Tuần 03-04

Web Control ASPX

Yêu cầu

- Nắm cơ bản ngôn ngữ C#
- Hiểu về lập trình hướng đối tượng
 - Thuộc tính?
 - Phương thức?
 - Sự kiện?

Một số thuộc tính

Tên thuộc tính	Ý nghĩa
AccessKey	Cho phép người dùng truy xuất nhanh đến control bằng phím tắt. Giả sử đặt giá trị là A, thì có thể truy xuất nhanh đến bằng cách sử dụng Alt+A
BackColor	Màu nền
BorderColor	Màu viền
BorderStyle	Kiểu viền(Dashed, Dotted, Double, Groove, Ridge, Inset, Outset, Solid, None)
BorderWidth	Độ dày viền
CssClass	Đặt css cho control
Enabled	Vô hiệu control(hiện nhưng không thực hiện được)
Font	Phông chữ
ForeColor	Màu chữ
Height	Độ cao
TabIndex	Thứ tự tab
Tooltip	Hiển thị chỉ dẫn
Width	Độ rộng

Các control cơ bản

Control	HTML	Thuộc tính, Phương thức, Sự Kiện
<asp:Button>	<input type="submit"/> <input type="button"/>	Text, CausesValidation, PostBackUrl, ValidationGroup, sự kiện Click
<asp:CheckBox>	<input type="checkbox"/>	AutoPostBack, Checked, Text, TextAlign, sự kiện CheckedChanged
<asp:FileUpload>	<input type="file"/>	FileBytes, FileContent, FileName, HasFile, PostedFile, SaveAs()
<asp:HiddenField>	<input type="hidden"/>	Value
<asp:HyperLink>	<a>...	ImageUrl, NavigateUrl, Target, Text
<asp:Image>		AlternateText, ImageAlign, ImageUrl

Các control cơ bản

Control	HTML	Thuộc tính, Phương thức, Sự Kiện
<asp:ImageButton>	<input type="image"/>	CausesValidation, ValidationGroup, sự kiện Click
<asp:ImageMap>	<map>	HotSpotMode, HotSpots (collection), AlternateText, ImageAlign, ImageUrl
<asp:Label>	...	Text, AssociatedControlID
<asp:LinkButton>	<a>	Text, CausesValidation, ValidationGroup, sự kiện Click
<asp:Panel>	<div>...</div>	BackImageUrl, DefaultButton, GroupingText, HorizontalAlign, Scrollbars, Wrap
<asp:RadioButton>	<input type="radio"/>	AutoPostBack, Checked, GroupName, Text, TextAlign, sự kiện CheckedChanged

Các control cơ bản

Control	HTML	Thuộc tính, Phương thức, Sự Kiện
<asp:Table>	<table>...	BackImageUrl, CellPadding, CellSpacing, GridLines, HorizontalAlign, Rows (mảng)
<asp:TableCell>	<td>...	ColumnSpan, HorizontalAlign, RowSpan, Text, VerticalAlign, Wrap
<asp:TableRow>	<tr>...</tr>	Cells (Mảng), HorizontalAlign, VerticalAlign
<asp:TextBox>	<input type="text"/> <textarea>...</textarea>	AutoPostBack, Columns, MaxLength, ReadOnly, Rows, Text, TextMode, Wrap, sự kiện TextChanged

Lớp Units

- Đặt độ đo cho các control

Ví dụ:

```
<asp:Panel Height="300px" Width="50%" id="pnl" runat="server" />
```



```
pnl.Height = Unit.Pixel(300);  
pnl.Width = Unit.Percentage(50);
```



```
Unit myUnit = new Unit(300, UnitType.Pixel);  
pnl.Height = myUnit;
```

Enumerations-Color

```
ctrl.BorderStyle = BorderStyle.Dashed;  
int alpha = 255, red = 0, green = 255, blue = 0;  
ctrl.ForeColor = Color.FromArgb(alpha, red, green, blue);  
ctrl.ForeColor = Color.Crimson;  
ctrl.ForeColor = ColorTranslator.FromHtml("Blue");
```


Lớp Font

Thuộc tính:

- Name
- Names
- Size
- Bold, Italic, Strikeout, Underline, Overline

Ví dụ:

```
ctrl.Font.Name = "Verdana";  
ctrl.Font.Bold = true;  
ctrl.Font.Size = FontUnit.Small;  
ctrl.Font.Size = FontUnit.Point(14);
```

Default Button

```
<form id="Form1" DefaultButton="cmdSubmit"  
runat="server">
```

Scrollable Panels

```
<asp:Panel ID="Panel1" runat="server" Height="116px"
Width="278px"
BorderStyle="Solid" BorderWidth="1px"
ScrollBars="Auto">
This scrolls.
<br /><br />
<asp:Button ID="Button1" runat="server"
Text="Button" />
<asp:Button ID="Button2" runat="server"
Text="Button" />
<br />
...
</asp:Panel>
```

Sự kiện-Phương thức Sự Kiện

```
<form id="form1" runat="server">
<div>
<h3>List of events:</h3>
<asp:ListBox id="lstEvents" runat="server" Height="107px" Width="355px"/>
<br /><br />
<h3>Controls being monitored for change events:</h3>
<asp:TextBox id="txt" runat="server" AutoPostBack="true"
OnTextChanged="CtrlChanged"/>
<br /><br />
<asp:CheckBox id="chk" runat="server" AutoPostBack="true"
OnCheckedChanged="CtrlChanged"/>
<br /><br />
<asp:RadioButton id="opt1" runat="server" GroupName="Sample"
AutoPostBack="true" OnCheckedChanged="CtrlChanged"/>
<asp:RadioButton id="opt2" runat="server" GroupName="Sample"
AutoPostBack="true" OnCheckedChanged="CtrlChanged"/>
</div>
</form>
```

Sự kiện

```
protected void CtrlChanged(Object sender, EventArgs e)
{
    string ctrlName = ((Control)sender).ID;
    lstEvents.Items.Add(ctrlName + " Changed");
    lstEvents.SelectedIndex = lstEvents.Items.Count - 1;
}
```

Sự kiện Click và ImageButton Control

```
protected void ImageButton1_Click(object sender,
System.Web.UI.ImageClickEventArgs e)
{
    lblResult.Text = "You clicked at (" + e.X.ToString() +
    ", " + e.Y.ToString() + "). ";
    // Kiểm tra vị trí click chuột
    // nếu nằm trong (20,20) đến (275,100) là bề mặt button.
    if ((e.Y < 100) && (e.Y > 20) && (e.X > 20) && (e.X < 275))
    {
        lblResult.Text += "You clicked on the button surface.";
    }
    else
    {
        lblResult.Text += "You clicked the button border.";
    }
}
```

List controls

Control	Mô tả
<asp:DropDownList>	Gồm một danh sách đổ xuống các đối tượng <asp:ListItem>. Được render thành <select> với size= «1»
<asp:ListBox>	Gồm một danh sách các đối tượng <asp:ListItem>. Được render thành <select> với size= «x» với x là số item
<asp:CheckBoxList>	Mỗi item được render thành checkbox, sắp trong một hay nhiều cột của table
<asp:RadioButtonList>	Tương tự <asp:CheckBoxList> nhưng mỗi item là một radio button
<asp:BulletedList>	Bullete, render thành

List controls

Thuộc tính	Mô tả
AutoPostBack	Tự động PostBack lại trang
Items	Danh sách các item
SelectedIndex	Chỉ số item đang được chọn. Trong trường hợp danh sách nhiều lựa chọn thì kiểm tra thuộc tính Selected ở từng Item
SelectedItem	Item đang được chọn. Trong trường hợp danh sách nhiều lựa chọn thì kiểm tra thuộc tính Selected ở từng Item
DataSource	Dữ Liệu nguồn hiển thị trên danh sách(DataTable, DataSet hay một danh sách đối tượng nào đó). Sử dụng phương thức DataBind() để điền DL vào danh sách

List controls

Thuộc tính	Mô tả
DataMember	Khi dữ liệu gồm nhiều Table(DataSet) thì thuộc tính này xác định table nào được chọn
DataTextField	Trường nào trong bảng được hiển thị ra màn hình
DataValueField	Trường nào trong bảng sử dụng để lấy giá trị
DataTextFormatString	Định dạng chuỗi hiển thị

List controls- *Ví dụ selectable controls*

Thuộc tính	Mô tả
RepeatLayout	Bao gồm các giá trị Table, Flow, UnorderedList, OrderedList để xác định checkbox hay radio button hiển thị theo table, trên cùng dòng, không có thứ tự hay có thứ tự.
RepeatDirection	Chiều hiển thị là ngang(Horizontal) hay dọc(Vertical)
RepeatColumns	Số cột, sử dụng khi RepeatLayout là Table
CellPadding, CellSpacing, TextAlign	Khoảng cách, canh lề khi RepeatLayout là Table

List controls- *Ví dụ selectable controls*

```
<form id="form1" runat="server">
<div>
<asp:ListBox runat="server" ID="Listbox1"
SelectionMode="Multiple" Rows="5">
<asp:ListItem Selected="true">Option 1</asp:ListItem>
<asp:ListItem>Option 2</asp:ListItem>
</asp:ListBox>
<br /><br />
<asp:DropDownList runat="server" ID="DropDownList1">
<asp:ListItem Selected="true">Option 1</asp:ListItem>
<asp:ListItem>Option 2</asp:ListItem>
</asp:DropDownList>
<br /><br />
```

List controls- *Ví dụ selectable controls*

```
<asp:CheckBoxList runat="server" ID="CheckboxList1"
RepeatColumns="3" >
<asp:ListItem Selected="true">Option 1</asp:ListItem>
<asp:ListItem>Option 2</asp:ListItem>
</asp:CheckBoxList>
<br />
<asp:RadioButtonList runat="server" ID="RadiobuttonList1"
RepeatDirection="Horizontal" RepeatColumns="2">
<asp:ListItem Selected="true">Option 1</asp:ListItem>
<asp:ListItem>Option 2</asp:ListItem>
</asp:RadioButtonList>
<asp:Button id="Button1" runat="server" Text="Submit"
OnClick="Button1_Click"/>
</div>
</form>
```

List controls- *Ví dụ selectable controls*

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!Page.IsPostBack)
    {
        for (int i=3; i<=5; i++)
        {
            Listbox1.Items.Add("Option " + i.ToString());
            DropDownList1.Items.Add("Option " + i.ToString());
            CheckboxList1.Items.Add("Option " + i.ToString());
            RadiobuttonList1.Items.Add("Option " + i.ToString());
        }
    }
}
```

List controls- *Ví dụ selectable controls*

```
protected void Button1_Click(object sender, EventArgs e)
{
    Response.Write("<b>Selected items for Listbox1:</b><br />");
    foreach (ListItem li in Listbox1.Items)
    {
        if (li.Selected) Response.Write("- " + li.Text + "<br />");
    }
    Response.Write("<b>Selected item for DropDownList1:</b><br />");
    Response.Write("- " + DropDownList1.SelectedItem.Text + "<br />");
}
```

List controls- *Ví dụ selectable controls*

```
Response.Write("<b>Selected items for CheckboxList1:</b><br />");
foreach (ListItem li in CheckboxList1.Items)
{
    if (li.Selected) Response.Write("-" + li.Text + "<br />");
}
Response.Write("<b>Selected item for RadiobuttonList1:</b><br />");
Response.Write("-" + RadiobuttonList1.SelectedItem.Text + "<br />");
}
```

List controls- *Ví dụ BulletedList Controls*

Thuộc tính	Mô tả
BulletStyle	Kiểu bullete gồm: Numbered (1, 2, 3...) LowerAlpha (a, b, c...) UpperAlpha (A, B, C...) LowerRoman (i, ii, iii...) UpperRoman (I, II, III...) Disc hay Circle hay Square CustomImage (khi sử dụng cần gán thuộc tính BulletImageUrl)
BulletImageUrl	Hình ảnh cho Bullete (BulletStyle là CustomImage)
FirstBulletNumber	Chỉ số bắt đầu dành cho danh sách có thứ tự
DisplayMode	Chế độ hiển thị:(Text, LinkButton , HyperLink)

List controls- *Ví dụ BulletedList Controls*

```
<asp:BulletedList ID="BulletedList1" runat="server"
BulletStyle="Disc" DisplayMode="LinkButton"
OnClick="BulletedList1_Click">
    <asp:ListItem Selected="True"
Text="opt1"></asp:ListItem>
    <asp:ListItem Selected="True"
Text="opt1"></asp:ListItem>
    <asp:ListItem Selected="True"
Text="opt1"></asp:ListItem>
</asp:BulletedList>
```

List controls- *Ví dụ BulletedList Controls*

```
protected void BulletedList1_Click(object sender,
BulletedListEventArgs e)
{
    string itemText = BulletedList1.Items[e.Index].Text;
    Response.Write( "You choose item" + itemText);
}
```

Description	Value
Name:	<input type="text"/> *
ID (multiple of 5):	33 *
Day off 08/05/08-08/20/08	01/01/01 *
Age (>= 18):	5 *
E-mail:	m *
Password:	•• *
Re-enter Password:	• *

☒ Validators enabled
☒ Client-side validation enabled
☒ Show summary
☐ Show message box

Please review the following errors:

- Name is required
- ID must be a multiple of 5
- Day Off is not within the valid interval
- You must be at least 18-year-old
- E-mail is not in a valid format
- The passwords don't match

Input Validation Control

Control	Mô tả
<asp:RequiredFieldValidator>	Kiểm tra dữ liệu rỗng
<asp:RangeValidator>	Kiểm tra dữ liệu nằm trong khoảng giá trị nào
<asp:CompareValidator>	So sánh dữ liệu giữa 2 control hoặc control và một giá trị
<asp:RegularExpressionValidator>	Kiểm tra định dạng nhập dữ liệu
<asp:CustomValidator>	Do người dùng tự định nghĩa
<asp:ValidationSummary>	Thống kê các vi phạm nhập liệu

Input Validation Control

Thuộc tính	Mô tả
ControlToValidate	ID control cần kiểm tra
Display	Chế độ hiển thị lỗi: Static: cần tính toán trước kích thước để khi bị lỗi không vỡ layout (đặc biệt là với table) Dynamic: tự động tính toán, chuyển đổi
EnableClientScript	Cho phép kiểm tra DL ở client hay không, mặc định là true
Enabled	
ErrorMessage	Chuỗi lỗi xuất hiện khi sử dụng ValidationSummary control
Text	Hiển thị khi control nhập sai
IsValid	DL nhập đúng hay không

Input Validation Control

Thuộc tính	Mô tả
SetFocusOnError	Có giá trị là true, thì trình duyệt sẽ focus vào control nếu DL nhập sai
ValidationGroup	Cho phép gom nhiều ValidationControl thành một nhóm
Validate()	Kiểm lại giá trị IsValid(kiểm lại DL nhập control)

RequiredFieldValidator Control

```
<asp:TextBox runat="server" ID="Name" />  
<asp:RequiredFieldValidator runat="server"  
ControlToValidate="Name" ErrorMessage="Name is  
required"  
Display="dynamic">*  
</asp:RequiredFieldValidator>
```

RangeValidator Control

```
<asp:TextBox runat="server" ID="DayOff" />  
<asp:RangeValidator runat="server"  
Display="dynamic" ControlToValidate="DayOff"  
Type="Date" ErrorMessage="Day Off is not within  
the valid interval" MinimumValue="2013-08-05"  
MaximumValue="2013-08-20">*  
</asp:RangeValidator>
```

Lưu ý định dạng ngày tháng năm của WebServer.
Type: Currency, Date, Double, Integer, String.

CompareValidator Control

```
<asp:TextBox runat="server" ID="Age" />  
<asp:CompareValidator runat="server" Display="dynamic"  
ControlToValidate="Age" ValueToCompare="18"  
ErrorMessage="You must be at least 18 years old"  
Type="Integer" Operator="GreaterThanEqual">*  
</asp:CompareValidator>
```

```
<asp:TextBox runat="server" TextMode="Password" ID="Password" />  
<asp:TextBox runat="server" TextMode="Password" ID="Password2" />  
<asp:CompareValidator runat="server"  
ControlToValidate="Password2" ControlToCompare="Password"  
ErrorMessage="The passwords don't match"  
Type="String" Display="dynamic">  
  
</asp:CompareValidator>
```

Lưu ý:

Operator: Equal, NotEqual, GreaterThan, GreaterThanEqual, LessThan, LessThanEqual, DataTypeCheck

RegularExpression-Loại ký tự

- Những ký tự đặc biệt sử dụng trong Rex: `.$^{\[(|)*+?\\`
- `\b`: backspace
- `\t`: Tab
- `\n`: dòng mới
- `.`: Thể hiện cho bất kỳ ký tự nào trừ `\n`
- `[ueoai]`: Thể hiện cho 1 ký tự trong tập các ký tự cho phép.
- `[^ueoai]`: Thể hiện cho ký tự không nằm trong tập các ký tự cho phép.
- `[3-7a-dA-D]`: Thể hiện một khoảng các giá trị
- `\s`: thể hiện bất kỳ ký tự khoảng trắng nào(tab, space,...)
- `\S`: thể hiện bất kỳ ký tự không khoảng trắng nào
- `\d`: Thể hiện ký tự là các số
- `\D`: Thể hiện ký tự không là các số
- `\w`: các ký tự, ký số và dấu `_`
- `\W`: không là ký tự, ký số.

RegularExpression-Số lượng Ký tự

- *: 0 ký tự trở lên
- +: 1 Ký tự trở lên
- ?: 0 hay 1 ký tự
- {N}: N ký tự
- {N,}: $\geq N$ Ký tự
- {N,M}: từ N đến M ký tự

Ví dụ: [aeiou]{2,4}\+[1-5]

Có ý nghĩa gì?

Một số RegularExpression

Tên	REX
E-mail address	\S+@\S+\.\S+
Password	\w+
Specific-length password	\w{4,10}
Advanced password	[a-zA-Z]\w{3,9}
Another advanced password	[a-zA-Z]\w*\d+\w*
Limited-length field	\S{4,10}
Social Security number (US)	\d{3}-\d{2}-\d{4}

CustomValidator Control

```
<asp:TextBox runat="server" ID="EmpID" />  
<asp:CustomValidator runat="server"  
ControlToValidate="EmpID"  
ClientValidationFunction="EmpIDClientValidate"  
OnServerValidate="EmpIDServerValidate"  
ErrorMessage="ID must be a multiple of 5"  
Display="dynamic">*  
</asp:CustomValidator>
```

CustomValidator Control

```
<script type="text/javascript">  
function EmpIDClientValidate(ctl, args)  
{  
    // the value is a multiple of 5 if the modulus by 5 is 0  
    args.IsValid=(args.Value%5 == 0);  
}  
</script>
```

CustomValidator Control

```
protected void EmpIDServerValidate(object sender,
ServerValidateEventArgs args)
{
    try
    {
        args.IsValid = (int.Parse(args.Value)%5 == 0);
    }
    catch
    {
        // An error is most likely caused by non-numeric data.
        args.IsValid = false;
    }
}
```

ValidationSummary Control

```
<asp:ValidationSummary runat="server"  
ID="Summary" ShowSummary="true"  
ShowMessageBox="true" DisplayMode="BulletList"  
HeaderText="<b>Please review the following  
errors:</b>"  
>
```


Bật tắt Validator

```
foreach (BaseValidator validator in
Page.Validators)
{
    validator.Enabled = true;
    validator.EnableClientScript = true;
}
```

Validation Group

```
<form id="form1" runat="server">  
<div>  
<asp:Panel ID="Panel1" runat="server">  
<asp:TextBox ID="TextBox1" ValidationGroup="Group1"  
runat="server" />  
    <asp:RequiredFieldValidator ID="RequiredFieldValidator1"  
    ErrorMessage="*Required" ValidationGroup="Group1"  
    runat="server" ControlToValidate="TextBox1" />  
    <asp:Button ID="Button1" Text="Validate Group1"  
    ValidationGroup="Group1" runat="server" />  
</asp:Panel>  
<br />
```

Validation Group

```
<asp:Panel ID="Panel2" runat="server">  
<asp:TextBox ID="TextBox2" ValidationGroup="Group2"  
runat="server" />  
<asp:RequiredFieldValidator  
ID="RequiredFieldValidator2" ErrorMessage="*Required"  
ValidationGroup="Group2"  
ControlToValidate="TextBox2" runat="server" />  
<asp:Button ID="Button2" Text="Validate  
Group2" ValidationGroup="Group2" runat="server" />  
</asp:Panel>  
</div>  
</form>
```

File Edit View Favorites Tools Help Back Forward Stop Home

Address http://localhost/Chapter05/Validators.aspx

Description	Value
Name:	<input type="text"/> *
ID (multiple of 5):	33 *
Day off 08/05/08-08/20/08	01/01/01 *
Age (>= 18):	5 *
E-mail:	m *
Password:	••
Re-enter Password:	•

☒ Validators enabled
☒ Client-side validation enabled
☒ Show summary
☐ Show message box

Microsoft Internet Explorer

Please review the following errors:

- Name is required
- ID must be a multiple of 5
- Day Off is not within the valid interval
- You must be at least 18-year-old
- E-mail is not in a valid format
- The passwords don't match

OK

Please review the following errors:

- Name is required
- ID must be a multiple of 5
- Day Off is not within the valid interval
- You must be at least 18-year-old
- E-mail is not in a valid format
- The passwords don't match

Rich Control

- AdRotator
- Calendar
- MultiView, View, Wizard
- Substitution
- XML

Lưu ý

- Khi control nằm trong ContentPlaceHolder của master page thì ID lúc render ra sẽ không giống lúc đặt tên → lưu ý khi làm CSS, thao tác trên giao diện.



Tuần 05

Ứng dụng ASP.NET

Cấu trúc hoạt động ứng dụng ASP.NET

- Webserver
 - Cách hoạt động, xử lý yêu cầu?
 - Trong Visual Studio thì sao?
 - Virtual Directory (đường dẫn ảo)
 - Là gì?
 - Trong Visual Studio?
 - ASP.NET worker
- ([http://msdn.microsoft.com/en-us/library/ms225480\(v=vs.80\).aspx](http://msdn.microsoft.com/en-us/library/ms225480(v=vs.80).aspx))
- Application Domain
 - Là gì, do cái gì điều khiển?
 - Đặc điểm của mô hình này? Có khác gì so với các mô hình Webserver trước đó?
 - Vòng đời ứng dụng ASP.NET
 - Ứng dụng được tạo ra và hủy khi nào?
 - Khi thay đổi nội dung, cấu hình ứng dụng?
 - Web farm và Web garden?

Virtual Directory

Bao gồm:

- Web form (.aspx)
- Master pages (.master)
- Web services (.asmx)
- Code-behind files(.cs)
- web.config (cấu hình chế độ bảo mật, kiểm lỗi, quản lý trạng thái)
- global.asax (điều khiển sự kiện)
- Các file được biên dịch(dll) chứa các hàm cần sử dụng
- Các file tài nguyên của website:hình ảnh, css, XML,...

Cấu trúc thư mục ứng dụng

Thư mục	Mô tả
Bin	Chứa các file dll (các lớp mà Web sử dụng)
App_Code	Chứa các file mã nguồn(nếu dung VS không cần sử dụng)
App_GlobalResources	Chứa tài nguyên cho tất cả các page trong ứng dụng
App_LocalResources	Chứa tài nguyên cho các page được chỉ định
App_WebReferences	Chứa Webservice sử dụng (wsdl)
App_Data	Chứa file dữ liệu
App_Browsers	File cấu hình render trình duyệt(XML)
App_Themes	Chứa theme cho ứng dụng

Tập tin global.asax

- Cho phép *viết các trình điều khiển sự kiện cho các sự kiện hệ thống*, người dùng không thể truy xuất một cách trực tiếp, global.asax tự động trả lời các sự kiện cho ứng dụng.
- Cung cấp các dịch vụ như global.asa của asp truyền thống.
- *Viết mã cho global.asax tương tự như trong Webform nhưng không chứa các tag HTML hay ASP*

Global.asax-Ví dụ

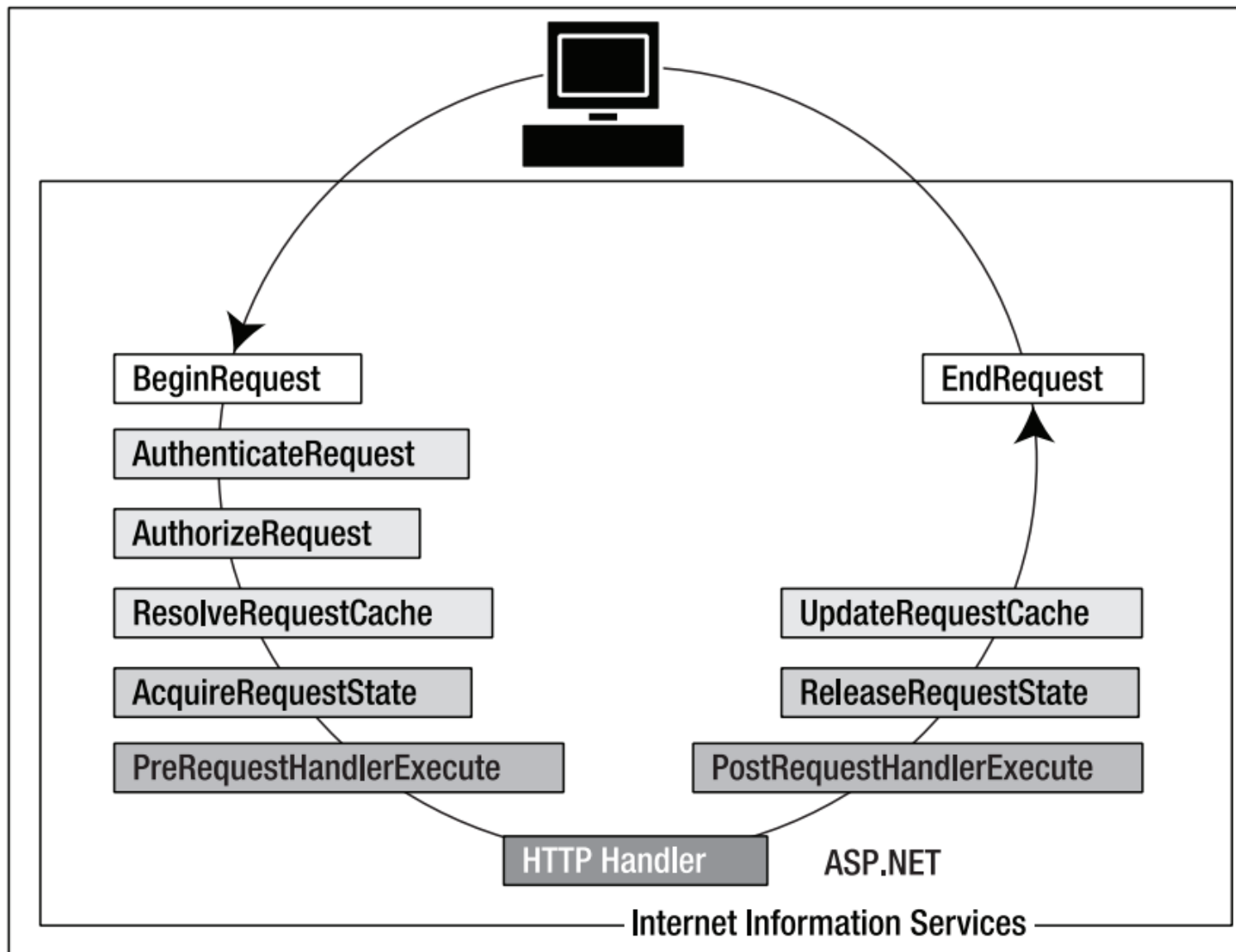
- Thêm file global.asax vào website
- Thêm đoạn code sau vào giữa cặp thẻ `<script></script>`.

```
protected void Application_OnEndRequest()  
{  
    Response.Write("<hr />This page  
was served at "+DateTime.Now.ToString());  
}
```

- Chạy xem kết quả

Nhận xét global.asax

- Tất cả file đều tạo phương thức cho lớp ứng dụng kế thừa từ `HttpApplication`, vì vậy có thể sử dụng các thuộc tính có tầm vực `Public` và `Protected`.
 - Ở trên đối tượng `Response` là thuộc tính của lớp
- Tương tác với các sự kiện:
 - `Application_BeginRequest()`: sự kiện được gọi ngay khi nhận được yêu cầu
 - `Application_AuthenticateRequest()`: sự kiện được gọi trước khi một chứng thực được thực thi
 - `Application_AuthorizeRequest()`: sau khi chứng thực người dùng, lúc bắt đầu xác định quyền của người dùng. Có thể dùng sự kiện này để phân các quyền đặc biệt.
 - `Application_ResolveRequestCache()`:Xảy ra trong quá trình lưu tạm kết quả lúc xử lý yêu cầu.
 - `Application_AcquireRequestState()`:trước khi thông tin `Session` được đưa ra cho client
 - `Application_PreRequestHandlerExecute()`:trước khi điều khiển giải quyết yêu cầu
 - `Application_PostRequestHandlerExecute()`:Sau khi yêu cầu được xử lý
 - `Application_ReleaseRequestState()`:Xảy ra khi session kế tiếp được gọi ra
 - `Application_UpdateRequestCache()`:trước khi thông tin được đưa vào cache
 - `Application_EndRequest()`:khi kết thúc yêu cầu



Các sự kiện khác của global.asax

- `Application_Start()`: khi một ứng dụng bắt đầu được khởi động và miền ứng dụng được tạo ra.
- `Session_Start()`: khi một session được tạo ra
- `Application_Error()`: lỗi ứng dụng xảy ra
- `Session_End()`: khi session bị loại bỏ
- `Application_End()`: trước khi ứng dụng kết thúc
- `Application_Disposed()`: khi ứng dụng bị tắt, cơ chế dọn rác của .NET hoạt động.

Ví dụ- Application_Error()

Global.asax (có thể viết vào sự kiện phát sinh sẵn trong *global.asax*)

```
<script language="C#" runat="server">
protected void Application_Error(Object sender, EventArgs e)
{
    Response.Write("<h1>");
    Response.Write("Co loi xay ra:</h1><hr />");
    Response.Write(Server.GetLastError().Message.ToString());
    Response.Write("<hr/>" + Server.GetLastError().ToString());
    Server.ClearError();
}
</script>
```


Ví dụ- Application_Error()

Default.aspx.cs

```
protected void Page_Load(object sender, EventArgs e)
{
    int i = 0;
    int j = 1;
    int k = j/i;
}
```

Lưu ý: Thử bỏ dòng `Server.ClearError()` và chạy lại, rút nhận xét

Cấu hình ứng dụng ASP.NET

- ASP.NET sử dụng các file XML để cấu hình
- Lợi ích của việc cấu hình:
 - Ứng dụng không bị khóa giữa chừng, trong suốt với người dùng
 - Truy cập chỉnh sửa từ xa qua FTP(chỉnh file xml)
 - Dễ dàng chỉnh sửa với cấu trúc file rõ ràng

machine.config

- Thường chứa trong thư mục
`c:\Windows\Microsoft.NET\Framework\[Version]\Config`
- Dùng để khai báo các cấu hình bổ sung(các section), cấu hình ASP.NET Worker process, đăng kí nhà cung cấp sử dụng các đặc tính nâng cao như tiểu sử(profile), kết bạn(membership), bảo mật.
- So với ASP.NET 1.x, có những khác biệt lớn về sau để tối ưu hóa các tiến trình

Computer > Win8 (C:) > Windows > Microsoft.NET > Framework64 > v4.0.30319 > Config

Name	Date modified	Type	Size
Browsers	26/07/2012 3:13 CH	File folder	
DefaultWsdllHelpGenerator.aspx	26/07/2012 3:11 CH	ASP.NET Server Pa...	69 KB
legacy.web_hightrust.config	26/07/2012 3:11 CH	XML Configuratio...	13 KB
legacy.web_hightrust.config.default	02/06/2012 9:33 CH	DEFAULT File	13 KB
legacy.web_lowtrust.config	26/07/2012 3:11 CH	XML Configuratio...	8 KB
legacy.web_lowtrust.config.default	02/06/2012 9:33 CH	DEFAULT File	8 KB
legacy.web_mediumtrust.config	26/07/2012 3:11 CH	XML Configuratio...	12 KB
legacy.web_mediumtrust.config.default	02/06/2012 9:33 CH	DEFAULT File	12 KB
legacy.web_minimaltrust.config	26/07/2012 3:11 CH	XML Configuratio...	7 KB
legacy.web_minimaltrust.config.default	02/06/2012 9:33 CH	DEFAULT File	7 KB
machine.config	22/12/2012 12:08 ...	XML Configuratio...	36 KB
machine.config.comments	26/07/2012 3:11 CH	COMMENTS File	90 KB
machine.config.default	26/07/2012 3:11 CH	DEFAULT File	36 KB
NetFx40_IIS_schema_update.xml	26/07/2012 3:11 CH	XML File	4 KB
NetFx45_IIS_schema_update.xml	26/07/2012 3:11 CH	XML File	3 KB
web.config	22/12/2012 12:05 ...	XML Configuratio...	43 KB
web.config.comments	26/07/2012 3:11 CH	COMMENTS File	68 KB
web.config.default	02/06/2012 9:33 CH	DEFAULT File	43 KB
web_hightrust.config	26/07/2012 3:11 CH	XML Configuratio...	11 KB
web_hightrust.config.default	02/06/2012 9:33 CH	DEFAULT File	11 KB
web_lowtrust.config	26/07/2012 3:11 CH	XML Configuratio...	6 KB
web_lowtrust.config.default	02/06/2012 9:33 CH	DEFAULT File	6 KB
web_mediumtrust.config	26/07/2012 3:11 CH	XML Configuratio...	10 KB
web_mediumtrust.config.default	02/06/2012 9:33 CH	DEFAULT File	10 KB
web_minimaltrust.config	26/07/2012 3:11 CH	XML Configuratio...	5 KB
web_minimaltrust.config.default	02/06/2012 9:33 CH	DEFAULT File	5 KB

<machineKey> trong machine.config

- Cấu hình mã hóa dữ liệu và tạo chữ ký điện tử
- Ví dụ:

```
<machineKey validationKey="AutoGenerate,IsolateApps"  
decryptionKey="AutoGenerate,IsolateApps" validation="SHA1" />
```

→ASP.NET sẽ tạo và lưu trữ 2 khóa xác định là machine-specific key và một khóa khác là application-specific key tự phát sinh cho mỗi ứng dụng. Điều này sẽ ngăn ngừa việc tấn công cross-site (tấn công dựa trên XSS)

Lưu ý:

- Nếu không sử dụng application-specific key thì bỏ đi *IsolateApps*.
- Với Web farm ứng dụng chạy trên nhiều máy chủ thì key phát sinh mỗi máy khác nhau nên ta cần cấu hình như sau:

```
<machineKey  
validationKey="61EA54E005915332011232149A2EEB317586824B265326CCDB3AD9  
ABDBE9D6F24B0625547769E835539AD3882D3DA88896EA531CC7AFE664866BD5242FC  
2B05D"decryptionKey="61EA54E005915332011232149A2EEB317586824B265337AF  
"validation="SHA1" />
```

Trong đó **validationKey** có độ dài từ 40-128 ký tự, **decryptionKey** 16(sử dụng DES) hoặc 48(sử dụng 3DES) ký tự. Nếu không đảm bảo thì server sẽ báo lỗi khi truy cập vào trang.

Web.config

- Tất cả ứng dụng đều kế thừa web.config và machine.config
- Tất cả cấu hình nằm trong thẻ <configuration>

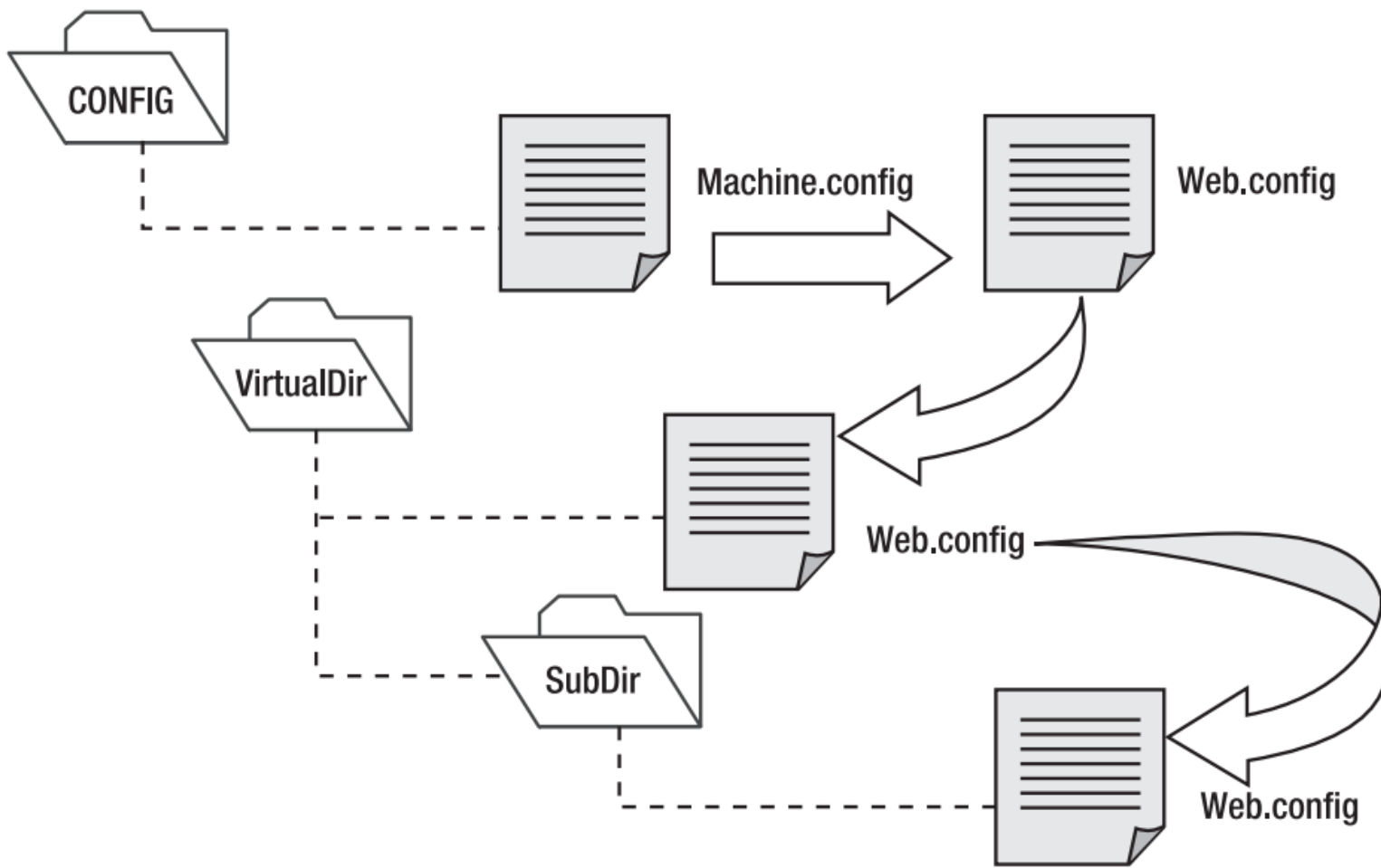
```
<?xml version="1.0"?>  
<configuration>  
  <appSettings />  
  <connectionStrings />  
  <system.web>  
    <!--ASP.NET configuration sections go here.-->  
  </system.web>  
  <system.webServer />  
</configuration>
```

Tính thừa kế trong cấu hình

Ví dụ: <http://localhost/A/B/C/MyPage.aspx>
trong đó A là thư mục chính thì các mức độ
kế thừa như sau:

1. machine.config
2. web.config của máy
3. web.config trong A
4. web.config trong B
5. web.config trong C

Tính thừa kế trong cấu hình



web.config - Thẻ <location>

- Cấu hình riêng cho một thư mục, file con

```
<configuration>
```

```
<system.web>
```

```
<!-- Basic configuration settings go here. -->
```

```
</system.web>
```

```
<location path="/Secure">
```

```
  <system.web>
```

```
    <!-- Configuration settings for the Secure  
    subdirectory go here. -->
```

```
  </system.web>
```

```
</location>
```

```
</configuration>
```

web.config-Thẻ <system.web>

- Cấu hình bảo mật, quản lý trạng thái, lưu vết
- Không thể thêm các thẻ con tự định nghĩa, các thẻ con trong <system.web> đã được quy định sẵn
- Các thẻ con: authentication, authorization, compilation, customErrors, membership, pages, profile, roleManager, sessionState, trace
- Phân biệt với thẻ <system.Web.Server> (thẻ này cấu hình cho server với các thẻ con <handlers>, <modules>)

<appSettings>

```
<?xml version="1.0" ?>
<configuration>
  <appSettings>
    <add key="websiteName" value="My New Website"/>
    <add key="welcomeMessage" value="Welcome to my new
Website, friend!"/>
  </appSettings>
  <system.web>...</system.web>
</configuration>
```

Truy xuất từ <appSettings>

```
using System.Web.Configuration;
```

```
protected void Page_Load(object sender, EventArgs e)  
{  
    lblSiteName.Text =  
        WebConfigurationManager.AppSettings["websiteName"];  
    lblWelcome.Text =  
        WebConfigurationManager.AppSettings["welcomeMessage"];  
}
```

Lưu ý: Tạo 2 label lblSiteName và lblWelcome

<connectionStrings>

```
<configuration>  
<connectionStrings>  
<add name="NorthwindConnection"  
connectionString=  
"Data Source=localhost;Integrated Security=SSPI;Initial  
Catalog=Northwind;"  
providerName="System.Data.SqlClient" />  
</connectionStrings>  
<system.web>...</system.web>  
</configuration>
```

Cách truy xuất:

```
string connectionString =  
WebConfigurationManager.ConnectionStrings["NorthwindConnection"]  
.Value;
```

Cài đặt-truy xuất thông tin cấu hình

- Sử dụng lớp WebConfigurationManager
- Các thuộc tính và phương thức:
 - AppSettings: truy xuất đến <appSettings>
 - ConnectionStrings: truy xuất đến <connectionStrings>
 - GetSection(): lấy về một đối tượng chứa thông tin cấu hình
 - OpenWebConfiguration(): trả về một đối tượng có thể truy cập đến việc cấu hình ứng dụng
 - OpenMachineConfiguration(): trả về một đối tượng có thể truy cập đến việc cấu hình server

- **Ví dụ:**

```
AuthenticationSection authSection =  
(AuthenticationSection)WebConfigurationManager.GetSe  
ction("system.web/authentication");
```

Cài đặt-truy xuất thông tin cấu hình

```
CompilationSection compSection =  
(CompilationSection)WebConfigurationManager.GetSection("s  
ystem.web/compilation");  
foreach (AssemblyInfo assm in compSection.Assemblies)  
{  
    Response.Write(assm.Assembly + "<br /");  
}
```

Website Administration Tool (WAT)

- Công cụ giúp cấu hình trên giao diện
- Vào Visual → chọn trên thanh menu Website(hoặc Project) → ASP.NET Configuration

Mở rộng cấu trúc file cấu hình

1. Xác định thông tin muốn lưu trữ trong file cấu hình sẽ tổ chức cấu trúc các thẻ và thuộc tính như thế nào? Nhiều thẻ có liên quan gom thành 1 nhóm và trong các thẻ có các thuộc tính thể hiện thông tin liên quan đến thẻ.
2. Với mỗi thẻ mới, tạo một lớp trong C# để lưu thông tin vào trong đối tượng của nó.
3. Các thẻ cấu hình mới phải chứa trong `<configSections>`

Tham khảo ví dụ trong tài liệu tham khảo

Mã hóa trong file cấu hình

Có 2 phương pháp mã hóa được hỗ trợ:

- RSA
- DPAPI

Bài tập

 ĐĂNG KÝ

Tên đăng nhập (*)

Mật khẩu (*)

Nhập lại mật khẩu(*)

Email

Số đăng ký

Địa chỉ

Ngày sinh

Ngày Tháng Năm

Nghề nghiệp

Sinh viên

Giáo viên

Công nhân

Kỹ sư

Khác

Phái

☒ Nam ☐ Nữ

Sở thích

☐ Thể thao ☐ Nghe nhạc ☐ Xem phim

Lưu

Làm mới

[Hướng dẫn thêm](#)

Yêu cầu

- Thiết kế Webform như trên, sử dụng css hợp lý(2 điểm)
 - Form đăng ký nằm giữa màn hình
 - Thêm màu sắc cho form
 - Xử lý khi click chuột vào textbox thì có màu vàng, di chuyển sang textbox khác thì trở lại bình thường
- Xử lý(7 điểm)
 - Kiểm tra dữ liệu: Tên đăng nhập, mật khẩu không được rỗng; mật khẩu và mật khẩu nhập lại phải giống nhau; mật khẩu phải vừa có chữ vừa có số; nhập đúng định dạng email; định dạng số đăng ký xxx-xxx-xxx trong đó x là một chữ số từ 1 đến 7.
 - Upload hình lên server
 - Xuất thông tin đăng ký phía dưới form đăng ký(định dạng xuất tự thiết kế phải đẹp)
- Nâng cao(2 điểm)
 - Khi kiểm tra dữ liệu nhập bị sai cho textbox hiện lên màu xanh nhạt.(có thể chọn màu khác tùy thích)
 - Xuất thông tin đăng ký sang một webpage khác(Nếu làm yêu cầu này không cần làm yêu cầu cuối cùng phần xử lý)
 - Cho hình ảnh hiện lên trước khi lưu



Tuần 06

.NET Component

Mục đích

- Xây dựng các lớp hỗ trợ xử lý
- Đóng gói
- Phân Tầng
- Dễ chỉnh sửa

Tạo một Component

- Sử dụng tạo các lớp (file .cs) trong App_Code.
- Sử dụng các dll từ Bin (*Các file dll này tạo từ project Class Library*)

Tạo component hỗ trợ đọc XML

Bước 1: Tạo thư mục XMLData, tạo file XMLfile.xml trong XMLData

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<note>
```

```
    <info alias="hehe">
```

```
        <name>Huỳnh Giàu Chắc</name>
```

```
        <place>Thành Phố Hồ Chí Minh</place>
```

```
        <detail>Nhà nghèo, con đông</detail>
```

```
        <status>Đói</status>
```

```
    </info>
```

```
    <info>
```

```
        <name>Trần Hên</name>
```

```
        <place>Hà Nội</place>
```

```
        <detail>Đẹp trai, Nhà Giàu</detail>
```

```
        <status>Dư tiền</status>
```

```
    </info>
```

```
</note>
```


Tạo component hỗ trợ đọc XML

Bước 2: Tạo lớp XMLReader và Quotation trong namespace XMLReader(trong file XMLReader.cs đặt trong thư mục App_Code)

Lớp Quotation gồm các thuộc tính đều có kiểu string:

- Họ Tên
- Nơi ở
- Chi tiết
- Trạng thái

Lớp XMLReader gồm các thuộc tính

- xmlDoc: XmlDocument (using System.xml)
- quoteCount:Số node (con của gốc)

Tạo component hỗ trợ đọc XML

Bước 3: Xây dựng phương thức khởi tạo (constructor) từ một XmlNode:

```
public Quotation(XmlNode quoteNode)
{
    if ((quoteNode.SelectSingleNode("name")) != null)
        _sHoTen = quoteNode.SelectSingleNode("name").InnerText;
    if ((quoteNode.Attributes.GetNamedItem("alias")) != null)
        _sHoTen =
        _sHoTen+"("+quoteNode.Attributes.GetNamedItem("alias").Value+")";
    if ((quoteNode.SelectSingleNode("place")) != null)
        _sNoiO = quoteNode.SelectSingleNode("place").InnerText;
    if ((quoteNode.SelectSingleNode("detail")) != null)
        _sChiTiet =
        quoteNode.SelectSingleNode("detail").InnerText;
    if ((quoteNode.SelectSingleNode("status")) != null)
        _sTinhTrang =
        quoteNode.SelectSingleNode("status").InnerText;
}
```

Tạo component hỗ trợ đọc XML

Bước 4: Xây dựng phương thức khởi tạo cho lớp XMLReader:

```
public XMLReader(string fileName)
{
    quoteDoc = new XmlDocument();
    quoteDoc.Load(fileName);
    quoteCount =
    quoteDoc.DocumentElement.ChildNodes.Count;
}
```

Tạo component hỗ trợ đọc XML

Bước 5: Xây dựng phương thức lấy 1 node ngẫu nhiên cho lớp XMLReader:

```
public Quotation GetRandomQuote()  
{  
    int i;  
    Random x = new Random();  
    i = x.Next(quoteCount - 1);  
    return new Quotation(quoteDoc.DocumentElement.ChildNodes[i]);  
}
```

Tạo component hỗ trợ đọc XML

Bước 6: Xây dựng phương thức lấy 1 node thứ i cho lớp XMLReader:

```
public Quotation GetQuote(int index)
{
    if (index < quoteCount)
    {
        return new Quotation(quoteDoc.DocumentElement.ChildNodes[index]);
    }
    return null;
}
```

Tạo component hỗ trợ đọc XML

Bước 7: Sử dụng các lớp vừa xây dựng

1. Tạo trang default.aspx (không kéo control vào)
2. Trong phương thức xử lý sự kiện PageLoad trang default.aspx.cs chèn vào: *(lưu ý using XMLReader;)*

```
protected void Page_Load(object sender, EventArgs e)
{
    XMLReader quotes = new
XMLReader(Server.MapPath("./XMLData/XMLFile.xml"));
    Quotation quote = quotes.GetQuote(0);
    Response.Write(quotes.QuoteCount.ToString()+"-<b>" +
quote.SHoTen + "</b> <br><i>" + quote.SChiTiet + "</i>-");
    Response.Write("<blockquote>" + quote.STinhTrang +
"</blockquote>");
}
```

Tạo DLL component

- Tạo project Class Library
 - Tạo file DLL và thêm vào thư mục Bin

HTTP Pipeline

- Đường truyền xử lý các sự kiện không giới hạn cho các webform
- tự định nghĩa các trình điều khiển sự kiện
- Sử dụng web.config để khai báo các trình điều khiển

HTTP Handler

```
<httpHandlers>  
<add verb="*" path="trace.axd" validate="true"  
type="System.Web.Handlers.TraceHandler"/>  
<add verb="*" path="*.config" validate="true"  
type="System.Web.HttpForbiddenHandler"/>  
<add verb="*" path="*.cs" validate="true"  
type="System.Web.HttpForbiddenHandler"/>  
<add verb="*" path="*.aspx" validate="true"  
type="System.Web.UI.PageHandlerFactory"/>  
</httpHandlers>
```



Các yêu cầu trang trace.axd được điều khiển bởi TraceHandler

Custom HTTP Handler

- IHttpHandler
 - ProcessRequest(): được gọi khi yêu cầu được nhận
 - IsReusable: thuộc tính kiểm tra HTTP handler có được tiếp tục sử dụng hay không.

Sử dụng FreeTextBox Editor

- Chuẩn bị: FTBv3-3-1.rar (download từ Website <http://freetextbox.com/download/>)
 - Docs: Thư mục chứa tài liệu giới thiệu về freetextbox
 - Framework-x.x: file dll freetextbox để thêm vào Website.
 - readme.txt: hướng dẫn cài đặt sử dụng
 - web.config: cấu hình cho Web chạy freetextbox
 - aspnet_client: Các file hình ảnh, js cần cho việc chạy freetextbox
 - ftb.imagegallery.aspx: Xử lý các thao tác thêm ảnh, quản lý ảnh

Sử dụng FreeTextBox Editor

- Bước 1: Thêm FreeTextBox.dll vào thư mục Bin (chọn đúng phiên bản .NET đang dùng)
- Bước 2: Cấu hình file web.config của Website (thêm handler cho các sự kiện của freetextbox), cần chú ý đến phiên bản .NET

Sử dụng FreeTextBox Editor

```
<?xml version="1.0"?>
<configuration>
  <system.web>
    <compilation debug="false" targetFramework="4.0" />
    <httpRuntime requestValidationMode="2.0" />
    <httpHandlers>
      <add verb="GET" path="FtbWebResource.axd"
type="FreeTextBoxControls.AssemblyResourceHandler,
FreeTextBox" />
    </httpHandlers>
  </system.web>
</configuration>
```

→ Phần httpRuntime dành cho .NET 4.0

Sử dụng FreeTextBox Editor

- Bước 3: Tạo trang Default.aspx (Chỉ gồm 1 FreeTextBox, 1 button và 1 label)
 - Khai báo sử dụng FTB (trên đầu trang .aspx, sau <%Page...>)

```
<%@ Register TagPrefix="FTB"  
Namespace="FreeTextBoxControls"  
Assembly="FreeTextBox" %>
```

- Sử dụng control FTB

```
<FTB:FreeTextBox id="FreeTextBox1"  
runat="Server" />
```

Sử dụng FreeTextBox Editor

- chỉnh sửa thêm thuộc tính cho FTB
1. ToolbarLayout(tạo thanh công cụ)
 - Các toolbar trong một loại cách nhau bởi dấu ,
 - Qua loại mới cách nhau bởi dấu ;
 - Qua vùng mới cách nhau bởi dấu |

Ví dụ:

```
ToolbarLayout="paragraphmenu,fontsizesmenu;bold,italic,underline|bulletedlist,numberedlist"
```



Sử dụng FreeTextBox Editor

Các giá trị thuộc tính ToolbarLayout:

Phiên bản miễn phí: ParagraphMenu, FontFacesMenu, FontSizesMenu, FontForeColorMenu,

FontForeColorPicker, FontBackColorMenu, FontBackColorPicker, Bold, Italic, Underline, Strikethrough, Superscript, Subscript, InsertImageFromGallery, CreateLink, Unlink, RemoveFormat, JustifyLeft, JustifyRight, JustifyCenter, JustifyFull, BulletedList, NumberedList, Indent, Outdent, Cut, Copy, Paste, Delete, Undo, Redo, Print, Save, ieSpellCheck, StyleMenu, SymbolsMenu, InsertHtmlMenu, InsertRule, InsertDate, InsertTime, WordClean, InsertImage, InsertTable, EditTable, InsertTableRowBefore, InsertTableRowAfter, DeleteTableRow, InsertTableColumnBefore, InsertTableColumnAfter, DeleteTableColumn, InsertForm, InsertForm, InsertTextBox, InsertTextArea, InsertRadioButton, InsertCheckBox, InsertDropDownList, InsertButton, InsertDiv, InsertImageFromGallery, Preview, SelectAll, EditStyle

Phiên bản Pro (hoặc chỉ chạy trên local): FontForeColorPicker, FontBackColorPicker, EditTable

InsertTableRowAfter, DeleteTableRow, InsertTableColumnBefore, InsertTableColumnAfter, DeleteTableColumn, InsertForm, InsertForm, InsertTextBox, InsertTextArea, InsertRadioButton, InsertCheckBox, InsertDropDownList, InsertButton, InsertDiv, Preview, SelectAll, EditStyle, WordClean

2. Tự tạo Toolbar cho FreeTextBox

```
<FTB:FreeTextBox ID="FreeTextBox1" AutoGenerateToolbarsFromString="false" runat="server">
  <Toolbars>
    <FTB:Toolbar runat="server">
      <FTB:ParagraphMenu runat="server" />
      <FTB:FontSizesMenu runat="server" />
    </FTB:Toolbar>
    <FTB:Toolbar runat="server">
      <FTB:Bold runat="server" />
      <FTB:Italic runat="server" />
      <FTB:Underline runat="server" />
      <FTB:ToolbarSeparator runat="server" />
      <FTB:BulletedList runat="server" />
      <FTB:NumberedList runat="server" />
    </FTB:Toolbar>
    <FTB:Toolbar runat="server">
      <FTB:InsertHtmlMenu runat="server">
        <Items>
          <FTB:ToolbarListItem Text="Cool1" Value="<b>lalala</b>" runat="server" />
          <FTB:ToolbarListItem Text="Cool2" Value="<i>lalala</i>" runat="server" />
          <FTB:ToolbarListItem Text="Cool3" Value="<u>lalala</u>" runat="server" />
        </Items>
      </FTB:InsertHtmlMenu>
    </FTB:Toolbar>
  </Toolbars>
</FTB:FreeTextBox>
```

3. Tự tạo Toolbar cho FreeTextBox (sử dụng code có thể chèn vào thẻ `<script runat="server">` hoặc hàm `page_load` trong trang .cs)

`<script runat="server">`

```
void Page_Load(object Src, EventArgs E) {  
    Toolbar toolbar1 = new Toolbar();  
    toolbar1.Items.Add(new ParagraphMenu());  
    toolbar1.Items.Add(new FontSizesMenu());  
  
    FreeTextBox1.Toolbars.Add(toolbar1);  
}
```

```
Toolbar toolbar2 = new Toolbar();  
toolbar2.Items.Add(new Bold());  
toolbar2.Items.Add(new Italic());  
toolbar2.Items.Add(new Underline());  
toolbar2.Items.Add(new ToolbarSeparator());  
toolbar2.Items.Add(new BulletedList());  
toolbar2.Items.Add(new NumberedList());
```

```
FreeTextBox1.Toolbars.Add(toolbar2);
```

```
Toolbar toolbar3 = new Toolbar();  
StyleMenu styleMenu = new StyleMenu();  
styleMenu.Items.Add(new ToolbarListItem("Highlight", "Highlight"));  
styleMenu.Items.Add(new ToolbarListItem("SmallCaps", "smallcaps"));
```

```
toolbar3.Items.Add(styleMenu);
```

```
FreeTextBox1.Toolbars.Add(toolbar3);
```

```
}
```

```
</script>
```

Sử dụng FreeTextBox Editor

Bước 4: Sử lý sự kiện click button lấy Text trong FTB xuất ra label.

```
Label1.Text = FreeTextBox1.Text;
```

- Nếu gặp lỗi thì xem trang Default.aspx thêm vào thuộc tính `ValidateRequest="false"` vào trong `<%Page....>`

Lưu ý: *Khi chèn vào trang .aspx có sử dụng masterpage đôi lúc sẽ bị lỗi về cách hiển thị giao diện (các control bỏ vào masterpage thì ID phát sinh không giống như ID gán ban đầu mà thêm vào nó của contentplaceholder nào), sửa bằng cách thêm css cho ID phát sinh này(chạy lên trình duyệt Chrome sử dụng chức năng kiểm tra phần tử)*



Tuần 7

Quản lý trạng thái

Một số thay đổi trong quản lý trạng thái ở ASP.NET 4

- Opt-in view state: cho phép disable ViewState
- Session Compression: Nén, làm giảm dữ liệu kho thao tác
- Selectively enabling session state: thêm phương thức
`HttpContext.SetSessionStateBehavior()`
- Partial session state: Khái niệm lưu trữ, truy xuất trạng thái từng phần, theo thứ tự.

ViewState

- Tạo trang Default1.aspx, Default2.aspx
 - Thêm vào 1 button đặt ID là btnDem, Text là Đếm, 1 Label ID là lblHienThi, Text là 0
 - Xử lý phương thức Page_Load của trang như sau:

```
if (ViewState["Dem"] != null)
{
    lblHienThi.Text = ViewState["Dem"].ToString();
}
```

ViewState

- Xử lý phương thức của sự kiện click button của 2 trang như sau:

```
if (ViewState["Dem"] != null)
{
    int iDem =
Convert.ToInt32(ViewState["Dem"].ToString());
    iDem++;
    ViewState["Dem"] = iDem;
}
```


ViewState

Cách chạy:

- Chạy trang Default1.aspx, đồng thời chạy trang Default2.aspx
- Click và xem kết quả ở hai trang
→ So sánh và nhận xét phạm vi ảnh hưởng của ViewState. (Trang 1 có hiệu ViewState của trang 2 hay không?)

QueryString

- Tạo trang gui.aspx gồm 1 textbox và 1 button.
- Tạo trang nhan.aspx gồm 1 label
- Mục tiêu: gửi giữ liệu nhập trong textbox từ trang gui.aspx sang trang nhan.aspx
- Xử lý:
 - Trang gửi: tạo phương thức cho sự kiện click vào button gửi như sau

```
protected void btnGui_Click(object sender, EventArgs e)
{
    Response.Redirect("Nhan.aspx?txt=" +
txtGui.Text);
}
```

QueryString

– Trang nhận: Xử lý phương thức Page_Load như sau

```
protected void Page_Load(object sender, EventArgs e)
```

```
{  
    if(Request.QueryString["txt"]!=null)  
    {  
        lblHienThi.Text=Request.QueryString["txt"].ToString();  
    }  
}
```

→Nắm được cách thao tác truyền và nhận QueryString. (Trang Web có thể truyền QueryString cho chính nó?)

→Truyền 2 giá trị qua QueryString?(Sử dụng dấu & để ngăn cách?)

URL Encoding

- Khi gửi dữ liệu theo QueryString thường có các ký tự đặc biệt như khoảng trắng, \$-_.+!*'(), hay những ký hiệu trùng với cú pháp yêu cầu như "&"

→ Sử dụng `Server.UrlEncode([GiaTriBien])` để mã hóa chúng và giải mã bằng `HttpServerUtility.UrlDecode()`

Ví dụ:

```
Response.Redirect("Nhan.aspx?txt=" +  
Server.UrlEncode(txtGui.Text));
```

Cross-Page Posting

- PostBack đến một trang Web khác (bình thường là tự PostBack)
- Đây cũng là một cách gửi thông tin đến các trang.
- Ví dụ: Tạo 2 trang CrossPage1.aspx và CrossPage2.aspx.

Trang CrossPage1.aspx gồm 1 textbox và 1 button như sau:

```
<asp:TextBox runat="server"  
ID="txtThongTin"> </asp:TextBox>  
<asp:Button runat="server" ID="btnGui"  
PostBackUrl="CrossPage2.aspx" Text="Gui" />
```

Cross-Page Posting

- Trang CrossPage2.aspx nhận thông tin từ CrossPage1.aspx và hiển thị lên label

```
protected void Page_Load(object sender, EventArgs e)
```

```
{  
    if (PreviousPage != null)  
    {  
        lblThongTin.Text = "Trang gửi:" +  
            PreviousPage.Header.Title;  
    }  
}
```

Có thể lấy thông tin từ trang gửi:

```
protected void Page_Load(object sender, EventArgs e)
```

```
{  
    CrossPage1 prevPage = PreviousPage as CrossPage1;  
    if (prevPage != null)  
    {  
        // lấy thông tin  
    }  
}
```

Lưu ý: Nếu xảy ra lỗi thì thêm vào `<%@ PreviousPageType VirtualPath="CrossPage1.aspx" %>` ở trang nhận.

Cross-Page Posting

Nhận xét: Trong ví dụ trên việc truyền dữ liệu thông qua thuộc tính `PostBackUrl` của `Button`, ngoài ra có thể sử dụng phương thức `Server.Transfer("CrossPage2.aspx", true);`

- Thuộc tính `IsPostBack` và `IsCrossPagePostBack` dùng để xác định trang đang được gọi `PostBack` hay `CrossPagePostBack` từ một trang khác. (`CrossPagePostBack` cũng là một `PostBack`)
- Trường hợp có sử dụng `Validator` ở trang `CrossPage1` cần chú ý kiểm tra DL ở `CrossPage2`, tránh trường hợp Client không hỗ trợ/không cho sử dụng script.

```
if (PreviousPage != null)
{
    if (!PreviousPage.IsValid)
    {
        // Display an error message or just do nothing.
        Response.Redirect(Request.UrlReferrer.AbsolutePath +
"?err=true");//gửi lại trang trước
    }
    else
    { ... }
}
```

Cookies

- Khởi tạo Cookies:

```
HttpCookie cookie = new HttpCookie("TenCookies");
```

- Gán giá trị cho Cookies:

```
cookie["TenCookies"] = <GiaTri>;
```

- Thêm Cookies vào client:

```
Response.Cookies.Add(cookie);
```

- Thời gian cho Cookies:

```
cookie.Expires = <Thời gian đích>;
```

- Lấy giá trị Cookies:

```
HttpCookie cookie = Request.Cookies["TenCookies"];
```

- Hủy Cookies:

```
cookie.Expires = DateTime.Now.AddDays(-1);
```

```
Response.Cookies.Add(cookie);
```


Session

- Được quản lí bởi ứng dụng với ID(120 bits) được phát sinh tự động

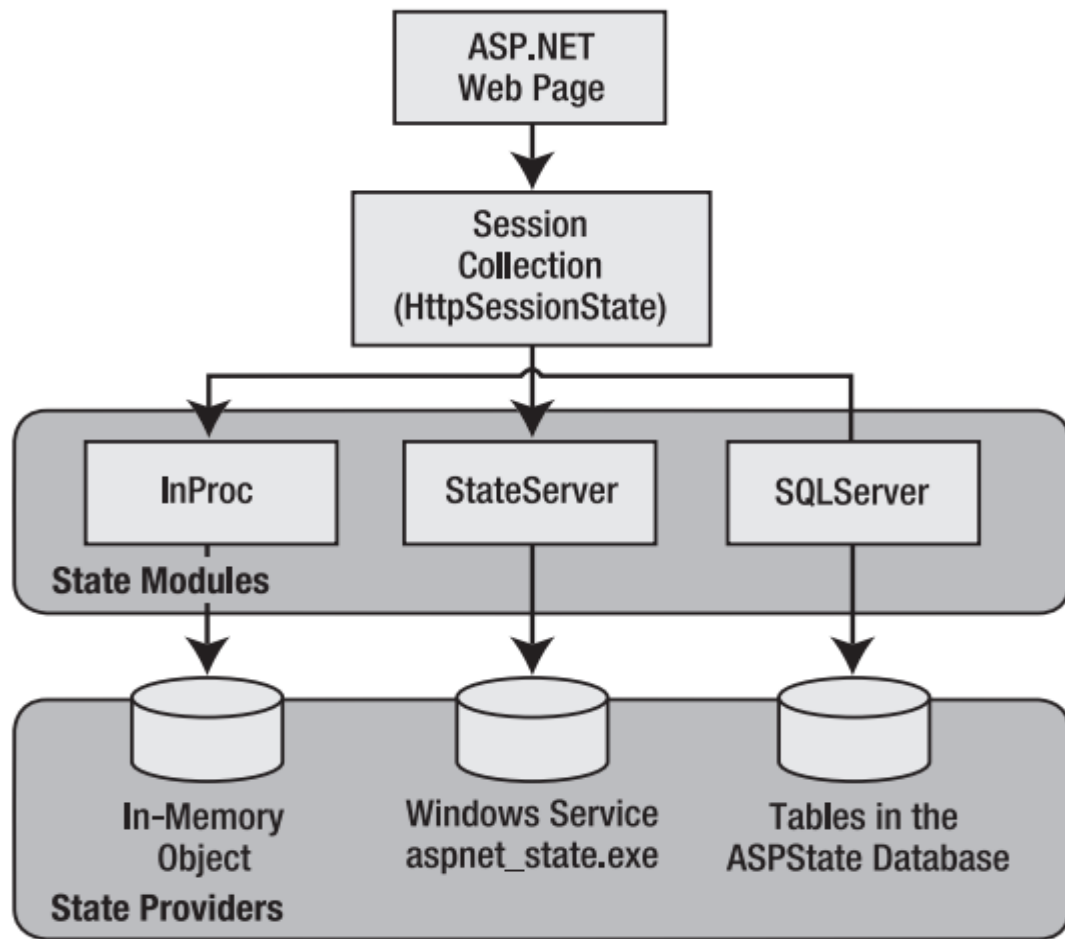


Figure 6-5. ASP.NET session state architecture

Sử dụng Session

- Gán giá trị:

```
Session["ProductsDataSet"] = dsProducts;
```

- Lấy giá trị

```
dsProducts =  
(DataSet)Session["ProductsDataSet"];
```

Mở rộng

- Cấu hình cho Session State-Các chế độ trong cấu hình trạng thái-Nén dữ liệu-Cookieless-Timeout-Securing Session State
→ Tham khảo sách:p261-268

HttpSessionState

- Thuộc tính:
 - Count: số lượng session
 - Timeout: Thời gian duy trì session
- Phương thức:
 - Abandon: hủy session hiện tại

Ví dụ Session

- Tạo ví dụ tương tự như ViewState(sửa ViewState → Session)

→Chạy thử 2 trang cùng lúc trên cùng 1 trình duyệt/khác trình duyệt, xem kết quả, nhận xét.(mỗi trình duyệt là 1 client)

(Session được hiểu ở bao nhiêu trang? Tại bao nhiêu client?)

Application

- Làm ví dụ tương tự như Session và so sánh kết quả

Static Application Variables

- Định nghĩa trong global.asax
- Ví dụ:

```
private static string[] fileList;  
public static string[] FileList  
{  
    get { return fileList; }  
}
```

- Lợi ích so với việc sử dụng Application
 - Tạo các mã chạy tự động
 - Lưu vết, kiểm tra DL

	ViewState	QueryString	CustomCookies
KDL cho phép	Các serializable .NET data(DL có cấu trúc tự định nghĩa, custom class)	Chuỗi	Chuỗi
Nơi lưu trữ	Là một trường ẩn trong 1 trang Web	Chuỗi URL của trình duyệt	Bộ nhớ máy client, tồn tại theo thời gian cài đặt trước
Lifetime(t/gian tồn tại)	Lúc trang Web được yêu cầu, chỉ dùng cho trang đó	Mất khi URL bị thay đổi	Được dùng ở nhiều trang, bị hủy khi hết thời gian cài trước
Scope(phạm vi)	Trang hiện tại	Trang đích(trên URL)	Chung cho tất cả
Security(bảo mật)	Được dùng mã chuỗi 64 bit để biểu diễn, dễ dàng thao tác đọc được.	Dễ thấy, dễ lấy	Không bảo mật, có thể thay đổi bởi người dùng
Hiệu suất	Nhiều làm chậm băng thông nhưng không ảnh hưởng đến hoạt động Webserver	Không ảnh hưởng gì vì DL ko nhiều	Không ảnh hưởng
Mục đích	Lưu thông tin 1 Trang	Gửi DL giữa các trang	Lưu thông tin người dùng tại client

	Session State	Application State
KDL cho phép	Các serializable .NET data	Các KDL .NET
Nơi lưu trữ	Ở server(mặc định)	Bộ nhớ ở server
Lifetime(t/gian tồn tại)	20 phút, hủy bằng Session.Abandon(), đóng trình duyệt	Mất khi URL bị thay đổi
Scope(phạm vi)	Các trang (cùng 1 client)	Các trang(tất cả client)
Security(bảo mật)	Bảo mật, tuy nhiên có thể bị hack nếu không sử dụng SSL	An toàn vì dữ liệu được lưu hoàn toàn trên Server
Hiệu suất	Nhiều làm chậm băng thông , chậm Webserver	Nhiều làm chậm server vì dữ liệu không bao giờ được xóa trừ khi tắt Webserver
Mục đích	Lưu giỏ hàng	DL toàn cục cho các trang

Profiles		Caching
KDL cho phép	Các serializable .NET data	Các KDL .NET, không hỗ trợ serializable data khi tạo custom profile
Nơi lưu trữ	CSDL cuối	Bộ nhớ ở server
Lifetime(t/gian tồn tại)	Lâu dài	Phụ thuộc vào chính sách cài đặt
Scope(phạm vi)	Tất cả các thành phần nằm trong ứng dụng,có thể truy xuất từ những ứng dụng mới	Các trang(tất cả client)
Security(bảo mật)	Tương đối vì DL ít phải chuyển tải, không được mã hóa	An toàn vì dữ liệu được lưu hoàn toàn trên Server
Hiệu suất	Có thể lưu dữ liệu lớn tuy nhiên ảnh hưởng đến tốc độ truy xuất, thêm thông tin vào DL	Nhiều làm quá tải
Mục đích	Lưu thông tin khách hàng	Lưu trữ DL truy xuất từ Database



Tuần 8

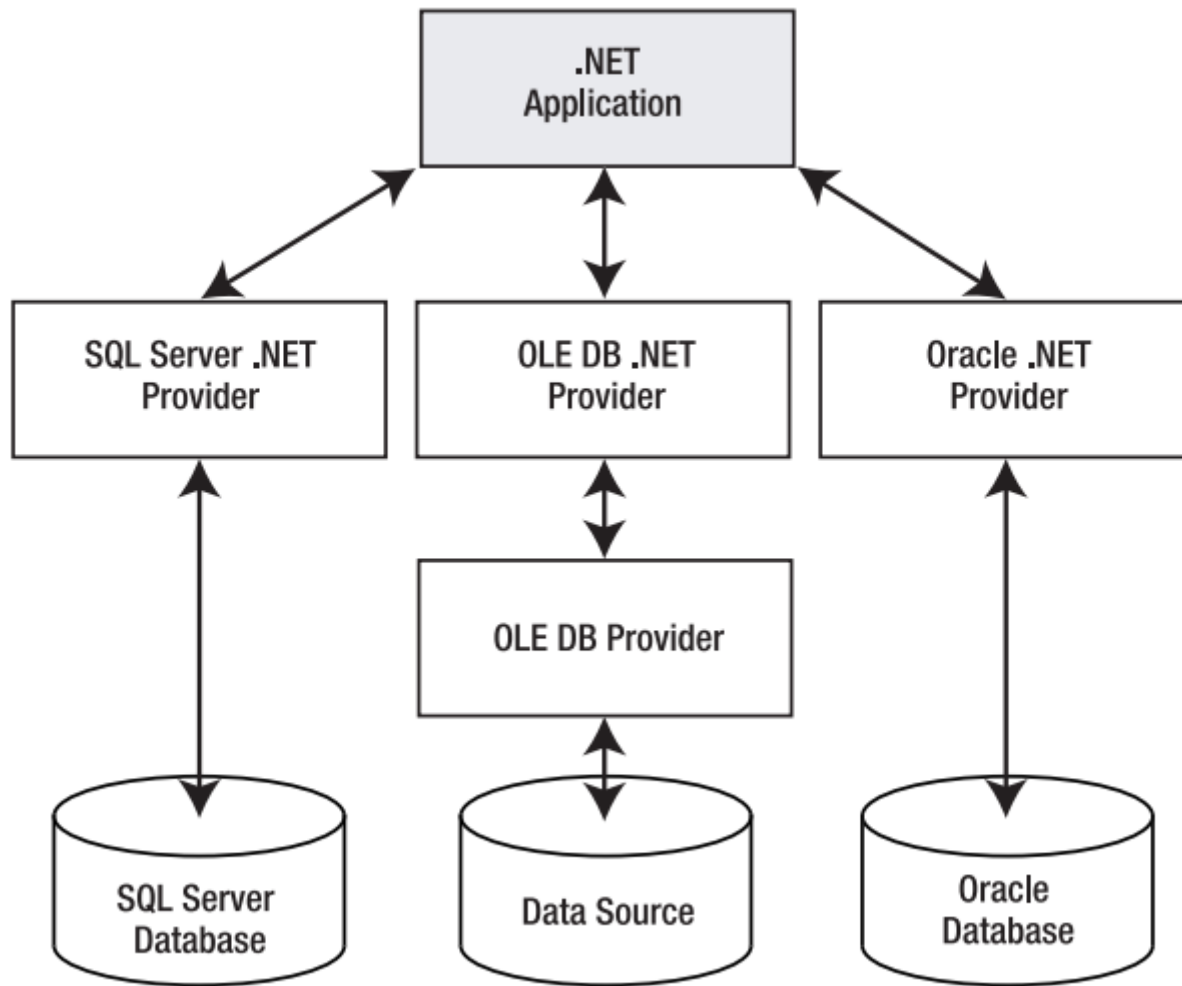
Thao tác với Dữ Liệu

Tổng quan ADO.NET

- Thuộc .NET Framework
- Kết nối dữ liệu, xử lý truy vấn, quản lý dữ liệu phi kết nối.
- Hỗ trợ cho cả ứng dụng desktop hoặc client-server

Các lớp trong ADO.NET

- Connection: quản lý kết nối
- Command: Thực hiện các câu lệnh SQL và thực thi các Stores Procedure
- DataReader: Đọc truy xuất dữ liệu
- DataAdapter; đưa DL vào dataset (gồm các bảng và quan hệ giữa chúng có thể làm việc trong tình trạng không kết nối), cập nhật thay đổi với dữ liệu nguồn
- Không có một lớp nào được sử dụng chung cho các hệ CSDL:
 - SqlConnection dùng cho SQL server
 - OleDbConnection dùng cho Access
- ADO.NET hỗ trợ 4 hệ CSDL:
 - SQL Server provider: phiên bản 7.0 về sau
 - OLE DB provider
 - Oracle provider
 - ODBC provider



Connection Classes- ConnectionString

- Chuỗi kết nối cần xác định:
 - Vị trí CSDL
 - Tên CSDL
 - Cơ chế chứng thực

Ví dụ câu kết nối SQL:

```
string connectionString = "Data Source=localhost; Initial
Catalog=Northwind;" +
"Integrated Security=SSPI";
string connectionString = "Data Source=localhost; Initial
Catalog=Northwind;" +
"user id=sa; password=opensesame";
```

Ví dụ với MS Access:

```
string connectionString =
"Provider=Microsoft.Jet.OLEDB.4.0;" +
@"Data Source=C:\DataSources\Northwind.mdb";
```

Connection Classes- ConnectionString

- Cấu hình trong Web.config

```
<configuration>
<connectionStrings>
<add name="Northwind" connectionString=
"Data Source=localhost; Initial Catalog=Northwind;
Integrated Security=SSPI"/>
</connectionStrings>
...
</configuration>
```

- Lấy giá trị từ Web.config:

```
string connectionString =
WebConfigurationManager.ConnectionStrings["Northwind
"].ConnectionString;
```




JavaScript

Client-Side Events

Event	Description	Applies To
onchange	Occurs when the user changes the value in an input control. In text controls, this event fires after the user changes focus to another control.	select, text, text area
onclick	Occurs when the user clicks a control.	button, check box, radio, link, area
onmouseover	Occurs when the user moves the mouse pointer over a control.	link, area
onmouseout	Occurs when the user moves the mouse pointer away from a control.	link, area
onkeydown	Occurs when the user presses a key.	text, text area
onkeyup	Occurs when the user releases a pressed key.	text, text area
onselect	Occurs when the user selects a portion of text in an input control.	text, text area
onfocus	Occurs when a control receives focus.	select, text, text area
onblur	Occurs when focus leaves a control.	select, text, text area
onabort	Occurs when the user cancels an image download.	image

Client-Side Events

Event	Description	Applies To
onerror	Occurs when an image can't be downloaded (probably because of an incorrect URL).	image
onload	Occurs when a new page finishes downloading.	window, location
onunload	Occurs when a page is unloaded. (This typically occurs after a new URL has been entered or a link has been clicked. It fires just before the new page is downloaded.)	window

Ví dụ

Tạo trang gồm form chứa 2 textbox, xử lý pageload như sau:

```
protected void Page_Load(object sender, EventArgs e)
{
    TextBox1.Attributes.Add("onmouseover",
    "alert('Your mouse is hovering on TextBox1.');"");
    TextBox2.Attributes.Add("onmouseover",
    "alert('Your mouse is hovering on TextBox2.');"");
}
```

→Chạy

The OnClientClick Property

```
<asp:Button id="btnClick" runat="server"  
OnClientClick="return confirm('Post back to the  
server?');" Text="Click Me"/>
```

