

Web Forms în ASP.NET

Radu Ionescu

Facultatea de Matematică și Informatică
Universitatea din București

raducu.ionescu@gmail.com

Cunoștințe necesare

Se presupune că următoarele concepte sunt deja cunoscute:

- Limbajul de programare C#
- HTML, DHTML
- JavaScript

Cunoștințe ce vor fi acumulate

Cursul cuprinde următoarele aspecte:

- Principalele caracteristici ale ASP.NET
- Web Forms

controale și evenimente, formulare, starea paginii, validarea datelor

Agendă

- [Arhitectura Web](#)
- Recapitulare ASP.NET
- Controale și evenimente
- Controale HTML
- Controale Web
- Controale de validare a datelor
- Concluzii

Arhitectura Web

Client
PC/Mac/Unix
+ Browser

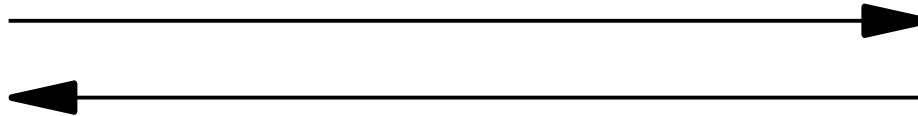


Rețea
HTTP, TCP/IP

Server Web



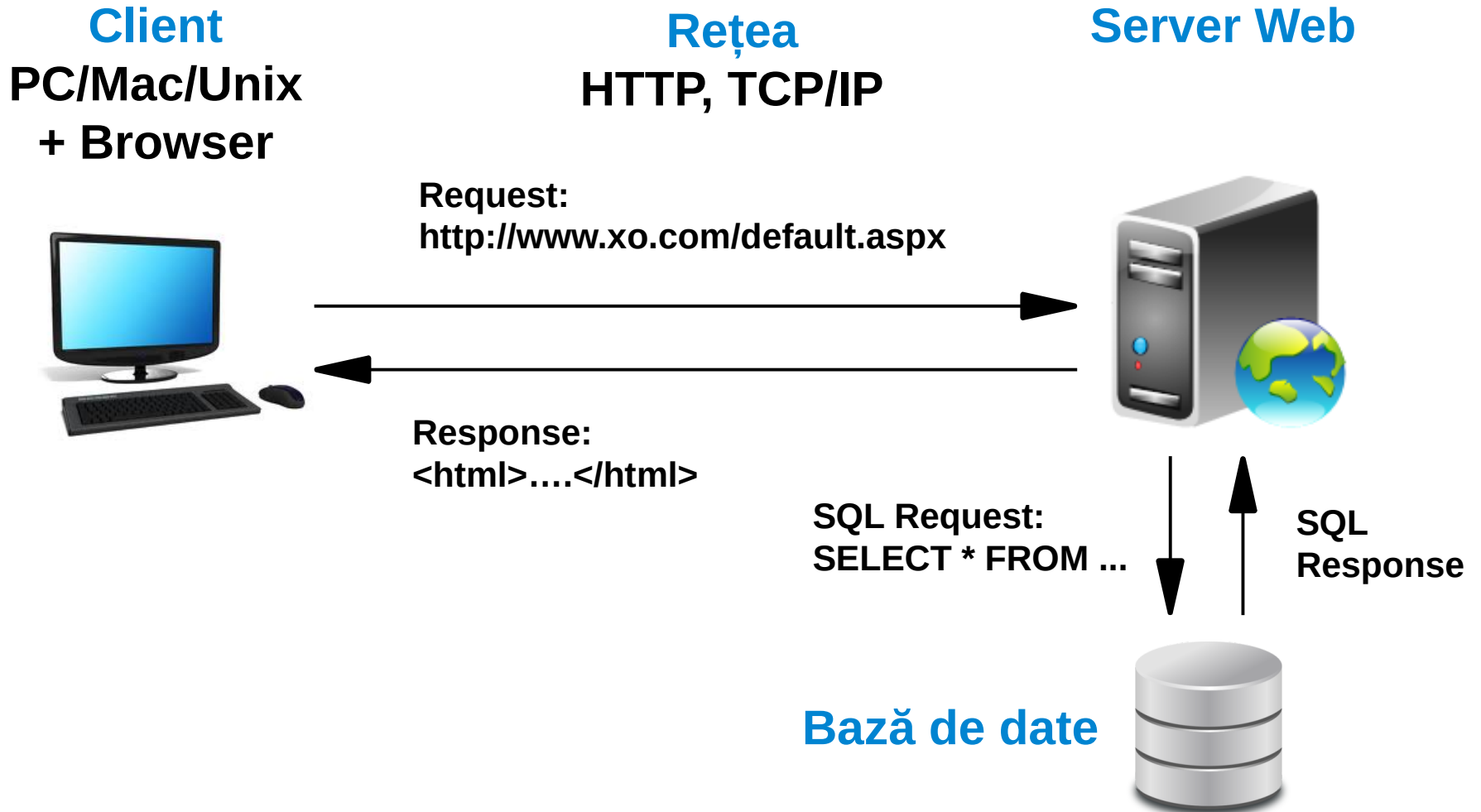
Request:
`http://www.xo.com/default.aspx`



Response:
`<html>....</html>`



Arhitectura Web



Tehnologii de dezvoltare

- La nivel de client:
HTML, DHTML, JavaScript
- La nivel de server:
ASP.NET (noua generație de ASP), C#

Agendă

- Arhitectura Web
- [Recapitulare ASP.NET](#)
- Controale și evenimente
- Controale HTML
- Controale Web
- Controale de validare a datelor
- Concluzii

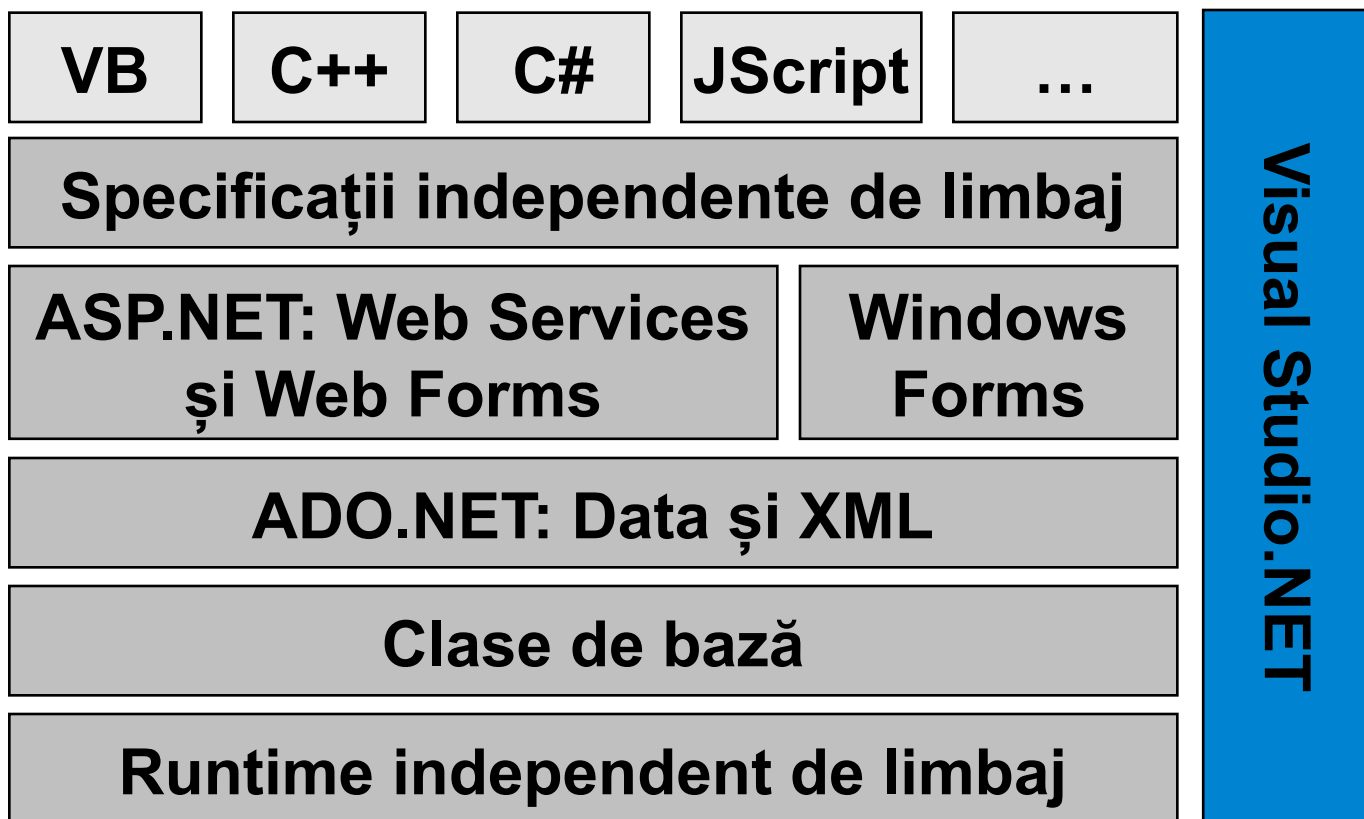
Concepte de bază în ASP.NET

- ASP.NET pune la dispoziție o serie de servicii și mecanisme pentru crearea, dezvoltarea și rularea aplicațiilor Web și serviciilor Web
- Aplicațiile Web sunt dezvoltate cu ajutorul Web Forms
- Web Forms au fost concepute astfel încât să permită o dezvoltare cât mai ușoară a aplicațiilor Web

Principalele caracteristici

- Web Forms
- Web Services
- .NET Framework
- Menține starea paginii
- Suport pentru mai multe browser-e Web
- Configurare prin XML
- Separarea codului de parte de interfață
- Managementul sesiunii
- Debugging
- Ușor de extins
- Securitate
- Modalitate simplă de validare a formularelor
- Sesiuni fără cookies

Arhitectura platformei ASP.NET



Hello World în ASP.NET

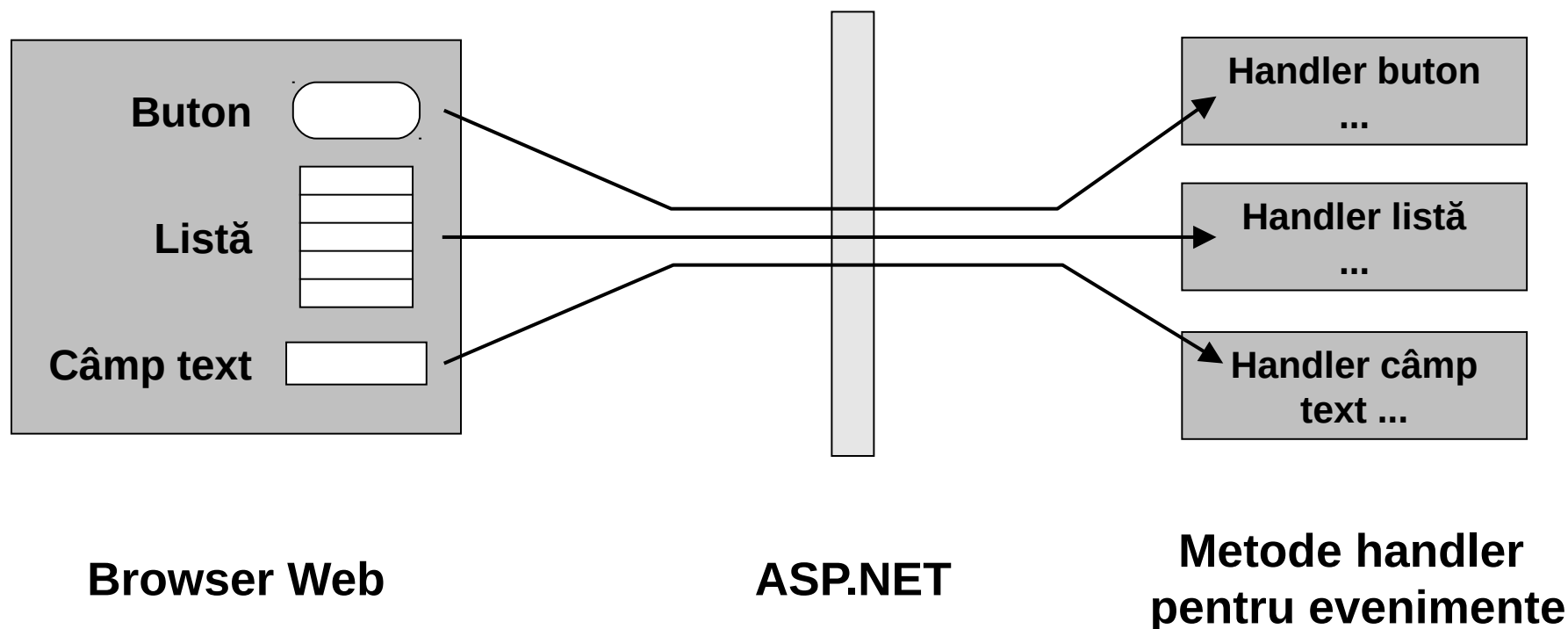
```
<%@ Page language="C#" %>
<html>
<head></head>
<script runat="server">
public void B_Click (object sender, System.EventArgs e) {
    Label1.Text = "Hello, the time is " + DateTime.Now;
}
</script>
<body>
<form method="post" runat="server">
<asp:Button onclick="B_Click" Text="Say Hello"
    runat="server" />
<p>
<asp:Label id=Label1 runat="server" />
</form>
</body>
</html>
```

Agendă

- Arhitectura Web
- Recapitulare ASP.NET
- **Controale și evenimente**
- Controale HTML
- Controale Web
- Controale de validare a datelor
- Concluzii

Modelul de programare

- Modelul de programare este bazate pe controale și evenimente care se execută la nivelul serverului



Pagini cu code-behind

- Există două stiluri de a dezvolta pagini ASP.NET:
 - Controalele și codul apar într-un singur fișier .aspx
 - Controalele apar în fișierul .aspx, iar codul într-un fișier separat (code-behind)
- Paginile cu code-behind permit separarea codului de partea de design a interfeței
- Programatorii și designerii pot lucra independent

Pagini cu code-behind

- Demo:
Behind.aspx
Behind.aspx.cs

Controale la nivel de server

- Controalele sunt obiecte care pot fi accesate și modificate din codul care rulează pe server
 - Corespund tag-urilor HTML, ASPX
 - Sunt derivate din `System.Web.UI.Control`
- Pagina Web este tot un obiect
 - Derivat din clasa `System.Web.UI.Page` care este derivată din `System.Web.UI.Control`

Controale la nivel de server

- Există o întreagă varietate de controale:
butoane, câmpuri text, liste, butoane radio, calendar, data grid, controale de validare
- Controalele pot fi:
 - Predefinite
 - Provenite din alte surse (3rd party)
 - Definite de utilizator
- Controalele pot fi populate prin data binding (cursul următor)

Sintaxa controalelor

- Controalele sunt declarate ca tag-uri HTML având atributul setat `runat="server"`

```
<input type="text" id="text3" runat="server" />  
<asp:Calendar id="myCal" runat="server" />
```

- Numele tag-ului identifică ce tip de control trebuie creat
- Atributul `id` oferă un identificator programatic
 - Numește instanța variabilei creată în urma unui eveniment postback (în mod asemănător cu Dynamic HTML)

Form-ul paginii

- Toate controalele de tip server trebuie să apară într-un tag `<form>`
- Tag-ul `<form>` trebuie să conțină atributul `runat="server"`
- Atributul `runat="server"` indică faptul că form-ul trebuie procesat la nivel de server.
- Se folosește `method="post"` sau `method="get"`
- O pagină .aspx poate conține un singur control de tip `<form runat="server">`

Proprietățile controalelor

- Atributele tag-urilor HTML sunt mapate prin proprietăți ale controalelor

```
<asp:Button id="c1" Text="Foo" runat="server">  
<asp:ListBox id="c2" Rows="5" runat="server">
```

- Tag-urile și atributele sunt case-insensitive
- Proprietățile controalelor pot fi setate programatic

```
c1.Text = "Foo";  
c2.Rows = 5;
```

Tipuri de controale

- La nivel logic, controalele sunt organizate în două familii:
 - Controale HTML
 - Controalele și proprietățile sunt mapate 1:1 cu tag-urile HTML
 - Controale Web
 - Funcționalitate îmbogățită
 - Oferă un model de obiecte consistent

Evenimente de tip postback

- Atunci când valorile dintr-un form HTML sunt trimise înapoi la sever (prin HTTP Post sau Get) se generează un eveniment postback
- Spre deosebire de alte tehnologii, ASP.NET menține în mod automat starea paginii în urma unui eveniment postback
- Obiectele corespunzătoare controalelor sunt populate automat în urma unui eveniment postback
- Starea paginii nu se păstrează pe server

Mentținerea stării paginii

- Controalele își mențin starea pe durata mai multor evenimente postback
 - Acest lucru este implementat folosind un input HTML de tip hidden: `__VIEWSTATE`
 - Funcționează pentru diverse tipuri de controale:
`TextBox, CheckBox, DropDownList, ListBox, Label, Literal, Calendar`
- Menținerea stării poate fi setată per control sau pentru întreaga pagină
 - Atributul `EnableViewState="true" / "false"`
 - Reduce dimensiunea variabilei `__VIEWSTATE`

Mentținerea stării paginii

- Demo:

State.asp

State.aspx

Evenimentele paginii

- Paginile sunt structurate folosind evenimente
 - Permite organizarea codului
- Se poate reacționa la evenimentele pagini
 - `Page_Load`
 - `Page_Unload`
- Se poate reacționa la evenimentele generate de controale
 - `Button1_Click`
 - `Textbox1_Changed`

Ciclul de viață al paginii

Inițializare —————

Page_Init

Restaurare stare controale —————

Încărcare pagină —————

Page_Load

Evenimente generate de controale

1. Evenimente de tip *Change* —

Textbox1_Changed

2. Evenimente de tip *Action* —

Button1_Click

Salvare stare —————

Randare pagină —————

Eliberare memorie —————

Page_Unload



Evenimente generate de controale

- Evenimente de tip *Change*
 - Acestea se execută la următorul eveniment de tip *Action*
 - Ex: `OnTextChanged`, `OnCheckedChanged`
 - Pentru a genera un postback trebuie setat atributul `AutoPostBack="true"`
 - Evenimentele de tip *Change* se pot executa într-o ordine aleatoare
- Evenimente de tip *Action*
 - Produc imediat un eveniment postback către server
 - Ex: `OnClick`
- Compatibile cu orice Browser Web

Atașarea metodelor handler la evenimente

- Metodele handler sunt identificate în tag

```
<asp:Button onclick="btn1_Click" runat=server>  
<asp:TextBox onchange="text1_changed" runat=server>
```

- Codul metodei handler

```
protected void btn1_Click(Object s, EventArgs e)  
{  
    Message.Text = "Button1 clicked";  
}
```

Agendă

- Arhitectura Web
- Recapitulare ASP.NET
- Controale și evenimente
- **Controale HTML**
- Controale Web
- Controale de validare a datelor
- Concluzii

Controale HTML

- Sunt ușor de folosit pentru designerii Web
- Proprietățile se mapează 1:1 cu atributele HTML:
`table.bgcolor = "red";`
- Utile atunci când trebuie convertite din pagini HTML existente
- Sunt derivate din
`System.Web.UI.HtmlControls.HtmlControl`
- Controalele suportate au o clasă specifică, iar celelalte sunt derivate din `HtmlGenericControl`

Controale HTML

Controale suportate:

- `<a>`
- ``
- `<form>`
- `<table>`
- `<tr>`
- `<td>`
- `<th>`
- `<select>`
- `<textarea>`
- `<button>`
- `<input type=text>`
- `<input type=file>`
- `<input type=submit>`
- `<input type=button>`
- `<input type=reset>`
- `<input type=hidden>`

Controale HTML

- Demo:
`HTMLControls.aspx`

Agendă

- Arhitectura Web
- Recapitulare ASP.NET
- Controale și evenimente
- Controale HTML
- **Controale Web**
- Controale de validare a datelor
- Concluzii

Controale Web

- Model de obiecte consistent

```
Label1.BackColor = Color.Red;  
Table2.BackColor = Color.Blue;
```

- Funcționalitate îmbogățită
 - Ex: `AutoPostBack`, metode adiționale
- Suport pentru diverse versiuni de Browser (uplevel și downlevel)
 - Ex: controalele de validare a datelor
- Toate controalele au un tip specific (strongly-typed)
 - Permite verificarea tipurilor la compilare

Controale Web

- Controalele Web apar în HTML sub formă de tag-uri care aparțin unui namespace
- Controalele Web au asociat prefixul `asp:`

```
<asp:Button onclick="button1_click" runat=server>  
<asp:TextBox onchanged="text1_changed" runat=server>
```

- Sunt definite în namespace-ul
`System.Web.UI.WebControls`
- Acest namespace se mapează automat pe prefixul
`asp:`

Controale Web

- Controalele Web oferă o serie de proprietăți pentru a controla modul de afișare și formatul:
 - Font
 - BackColor, ForeColor
 - BorderColor, BorderStyle, BorderWidth
 - Style, CssClass
 - Height, Width
 - Visible, Enabled

Tipuri de controale Web

- Sunt 4 tipuri de controale Web:
 - Controale intrinseci
 - Controale de tip listă (List)
 - Controale complexe (Rich)
 - Controale de validare a datelor

Controale intrinseci

- Corespund controalelor HTML
- Controale suportate:
 - `<asp:button>`
 - `<asp:imagebutton>`
 - `<asp:linkbutton>`
 - `<asp:hyperlink>`
 - `<asp:textbox>`
 - `<asp:checkbox>`
 - `<asp:radiobutton>`
 - `<asp:image>`
 - `<asp:label>`
 - `<asp:panel>`
 - `<asp:table>`

Controale intrinseci

- `TextBox`, `ListControl`, `CheckBox` și subclasele aferente nu generează un eveniment postback în mod automat în cazul unui eveniment de tip `Change`
- Pentru ca o schimbare să genereze imediat un eveniment postback, se setează atributul `AutoPostBack="true"`

Controale de tip listă

- Controalele de tip listă sunt utilizate pentru a prezenta date cu repetiție
- Controale suportate:
 - `<asp:DropDownList>`
 - `<asp:ListBox>`
 - `<asp:RadioButtonList>`
 - `<asp:CheckBoxList>`
 - `<asp:Repeater>`
 - `<asp:DataList>`
 - `<asp:DataGrid>`

Controale de tip listă

- Repeater, DataList, DataGrid
 - Controale customizabile care oferă posibilitatea dezvoltării unor pagini complexe într-un timp foarte scurt
 - Se bazează pe șabloane care pot fi modificate
 - Pot conține alte controale
 - Permit atașarea de metode handler pentru evenimente de tipul `OnItemCommand`
 - Aceste controale vor fi discutate în cursul viitor

Controale de tip listă

- CheckBoxList, RadioButtonList, DropDownList
- Permit afișarea unei colecții de date pentru selecție multiplă sau selecție exclusivă
- Pot fi populate prin data binding (cursul viitor)

```
<asp:DropDownList id=Check1 runat="server">  
  <asp:ListItem>Item 1</asp:ListItem>  
  <asp:ListItem>Item 2</asp:ListItem>  
  <asp:ListItem>Item 3</asp:ListItem>  
  <asp:ListItem>Item 4</asp:ListItem>  
  <asp:ListItem>Item 5</asp:ListItem>  
</asp:DropDownList>
```

Controale intrinseci și de tip listă

- Demo:
WebControls.aspx

Controale complexe

- Controale custom cu funcționalitate îmbogățită
- Controale suportate:
 - `<asp:calendar>`
 - `<asp:adrotator>`
- Alte controale de proveniență 3rd party

Agendă

- Arhitectura Web
- Recapitulare ASP.NET
- Controale și evenimente
- Controale HTML
- Controale Web
- Controale de validare a datelor
- Concluzii

Controale de validare a datelor

- Funcționalitate complexă folosind doar declarații (nu necesită cod)
- Validarea se declară separat de controalele de input
- Framework-ul de validare este extensibil
- Suportă validare atât la nivel de client cât și la nivel de server
 - Se evită trimiterea datelor nevalidate la server
- Validarea este oricum efectuată și pe server
 - Previne atacurile de tip spoofing

Controale de validare a datelor

- `<asp:RequiredFieldValidator>`
 - Asigură că a fost introdusă o valoare
- `<asp:RangeValidator>`
 - Verifică dacă o valoare se află între o valoare minimă și maximă (atenție la atributul `Type`)
 - Atribute specifice: `Type`, `MinimumValue`, `MaximumValue`
- `<asp:CompareValidator>`
 - Compară o valoare cu o constantă, cu un alt control sau cu un tip de date
 - Atribute specifice: `Operator`, `Type`, `ControlToCompare`, `ValueToCompare`,

Controale de validare a datelor

- `<asp:RegularExpressionValidator>`
 - Testează dacă o valoare este conformă cu o expresie regulată
 - **Attribute specifice:** `ValidationExpression`
- `<asp:CustomValidator>`
 - Permite adăugarea unei metode handler de validare (se poate executa la nivel de server sau la nivel de client)
 - **Attribute specifice:** `OnServerValidate`
- `<asp:ValidationSummary>`
 - Afişează o listă cu mesajele de eroare într-un singur loc

Controale de validare a datelor

- Controalele de validare conțin text care este afișat doar în cazul în care validarea eșuează
- Conținutul atributului `Text` este afișat în locația controlului de validare
- Conținutul atributului `ErrorMessage` este afișat în sumar

Controale de validare a datelor

- Pentru a asocia un control de validare unui control de input se setează proprietatea `ControlToValidate` a validatorului

```
<asp:TextBox id="TextBox1" runat="server" />  
  
<asp:RequiredFieldValidator id="Req1"  
    ControlToValidate="TextBox1"  
    Text="*"   
    ErrorMessage="Required Field" runat="server" />
```

- Se pot asocia mai multe controale de validare pentru un anumit control

Controale de validare a datelor

- `Page.IsValid` indică dacă toate datele au fost validate de către controalele de validare

```
void Submit_click(object s, EventArgs e)
{
    if (Page.IsValid)
    {
        Message.Text = "Page is valid!";
    }
}
```

Controale de validare a datelor

- Atributul `Display` controlează modul de layout
 - `Static`: layout fix, locația nu se schimbă în cazul nevalidării controlului asociat
 - `Dynamic`: layout dinamic
 - `None`: nu se afisează textul, dar se poate folosi în continuare `ValidationSummary` și `Page.IsValid`
- Atributul `Type` specifică tipul de date pentru un anumit câmp: `Currency`, `Date`, `Double`, `Integer`, `String`

Controale de validare a datelor

- Demo:
StudentForm.aspx

Agendă

- Arhitectura Web
- Recapitulare ASP.NET
- Controale și evenimente
- Controale HTML
- Controale Web
- Controale de validare a datelor
- Concluzii

Concluzii

- În acest curs am vorbit despre:
 - Principalele caracteristici ale ASP.NET
 - Web Forms
controale și evenimente, formulare, starea paginii, validarea datelor

Concluzii

- În cursul viitor vom acoperi:
 - Data Binding
 - Templates
 - Conexiuni SQL

Resurse

- Web Forms

<http://www.asp.net/web-forms>

http://www.w3schools.com/aspnet/aspnet_intro.asp

http://www.w3schools.com/aspnet/aspnet_forms.asp

<http://msdn.microsoft.com/en-us/library/ms973868.aspx>

- Validare

http://www.w3schools.com/aspnet/aspnet_refvalidationcontrols.asp