



Dynamische Webseiten mit ASP.NET

- Web Technologien von Microsoft
- Einfache dynamische Webseiten
- Web-Formulare
- Ereignisbehandlung
- Web-Controls
- Validatoren
- Werkzeuge (Visual Studio)



Microsoft Web Technologien

ASP



Vorgestellt 1996

Ziel: vereinfachte (im Vergleich zu CGI) Erstellung von funktionalen Web Seiten

Einbettung von VBS oder JavaScript Code innerhalb <% %> Tags,

- Wird auf dem Server ausgeführt;
- VBS und JS werden von der im Command-Shell unterstützt

Code wird interpretiert und nicht übersetzt -> inhärente Performanceprobleme

Komplexere Funktionalität in COM Objekte in C++ implementiert

```
<!DOCTYPE html>
<html>
<body>
<% response.write("Hello World!") %>
</body>
</html>
```

http://www.w3schools.com/asp/asp_syntax.asp

ASP.NET



- Vorgestellt 2002 (aktuelle Version 4.5, August 2012.)
 - Ziel: Entwicklung von Webanwendungen (konv. Formularverarbeitung)
 - Entwickler soll sich nicht um die Details von HTML, CSS, JavaScript, JSON, JQuery, Browser-Abhängigkeiten, Zustandsverwaltung, ... kümmern müssen.
- Soll auf einem gewohnten Paradigma (ähnlich der Windows Form) basieren
 - Hohes Abstraktionsniveau

```
<%@ Page Language="C#" Inherits="_HelloPage"
CodeFile="Hello.aspx.cs"%>
<html>
<body>
<form>
<asp:Label ID="L1" Text="Hello" Runat="server"/>
<form>
</body>
</html>
```

Merkmale von ASP.NET 2.0



- Ereignisorientiert
- Objektorientiert
- Interaktiv erstellbar (RAD)
- Reichhaltige Bibliothek von GUI-Elementen
- Selbstgeschriebene GUI-Elemente mögl.
- Trennung von Layout (HTML) und Logik (C#)
- Effizient (compilierte Server-Scripts)
- Zustandsverwaltung
- Autorisierung / Authentifizierung

🗿 http://www.ibuyspystore.com/ShoppingCart.aspx - Microsoft Internet Explorer - 🗆 × File Edit View Favorites Tools Help **Shopping Cart** Product Name Model Quantity Price Subtotal Product ID Remove Fake Moustache 401 TCKLR1 \$599.99 \$599.99 Translator 368 DNTRPR \$19,999.98 \$19,999.98 Toaster Boat Total: \$20,599.97 Update Your Shopping Cart Final Check Out 🔮 Internet 🙆 Done

...

MVC



- Vorgestellt 2008 als Machbarkeitsstudie
- jährlich neue Version (jetzt bereits 5)
- Ziel: Entwicklung von "vollwertigen" Webanwendungen
 - Aber: Entwickler darf sich wieder um die Details von HTML, CSS, JavaScript, JSON, JQuery, Browser-Abhängigkeiten, Zustandsverwaltung kümmern
- Webanwendung
 - Zu Desktop Anwendungen gleichwertig (z.B.OWA, Google Docs)
 - Aber meist funktional eingeschränkt
 - Datenspeicherung entfernt ist der Standard Fall
- Struktur der Anwendung basiert auf einem MVC Grundmuster
- MVC ist offizielle neue strategische Stossrichtung



http://www.codeproject.com/Articles/528117/WebForms-vs-MVC

To MVC or not To MVC, That's the <?/>



MVC	Web Forms
You want full control over markups	You like programming against the reusable control abstraction that encapsulate UI and logic
You want a framework that <i>enforces</i> separation of concerns	You like using the WYSWIG designer and would rather avoid angle brackets
TDD/Unit Testing is a priority for you	You like keeping logic on the server rather than hand writing JavaScript
Control abstractions get in your way more than they help	Unit testing with the MVP pattern is sufficient for your needs
You like writing JavaScript	
	(c) Noam King



Einfache dynamische Webseiten mit ASP.NET

Statische Webseiten



Reines HTML My.html <html> Request("My.html") <head> Server Browser (IIS) <title>Simple HTML page</title> Response(My.html) </head> <body> <h1>Welcome</h1> My.html You are visitor number 1! </body> </html> Simple HTML page - Microsoft I... 💶 🗖 🗙 Edit <u>V</u>iew Favorites Tool » File Welcome You are visitor number 1! 🔠 Local intranet 🞒 Do

School of Engineering

Dynamische ASPX-Seiten



```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.IO" %>
                                                   Füge auf Server berechnete Werte in
<html>
                                                   HTML-Code ein
   <head> <title>Page counter</title> </head>
  <body>
     <h1>Welcome</h1>
     You are visitor number <%
     FileStream s = new FileStream("c:\\Data\\Counter.dat",
                                        FileMode.OpenOrCreate);
     int n;
     try {
        BinaryReader r = new BinaryReader(s);
                                                           🖉 Page counter - Micro... 💶 🗖 🗙
        n = r.ReadInt32();
     } catch { n = 0; } // falls die Datei leer ist
                                                                 Edit View Favc 🌺
                                                             File
     n++;
     s.Seek(0, SeekOrigin.Begin);
                                                             Welcome
     BinaryWriter w = new BinaryWriter(s);
     w.Write(n); s.Close();
     Response.Write(n);
                                                            You are visitor number 6!
     %>!
   </body>
                                                                  📴 Local intranet
</html>
```

School of Engineering

Was geschieht hinter den Kulissen?





School of Engineering

Zurückgesandter HTML-Code



Counter.aspx

Zurückgesandter HTML-Code

```
<html>
<head><title>Page counter</title></head>
<body>
<h1>Welcome</h1>
You are visitor number 6 !
</body>
</html>
```

 Enthält keinen Script-Code mehr
 Jeder Browser kann dieses HTML anzeigen

Code in Script-Tags





Ohne Code Behind erzeugte Seitenklasse





Generierte Klasse Counter_aspx



```
namespace ASP {
  using System.IO;
  public class Counter_aspx : Page {
                                                                       C:\WINDOWS\Microsoft.NET\Framework
     private static bool initialized = false;
                                                                       \v2.0.50727\Temporary ASP.NET Files
     private static ArrayList __fileDependencies;
     public Counter_aspx() {
        ArrayList dependencies;
        if ((__initialized == false)) { ... }
     public override string TemplateSourceDirectory {
        get { return "/Samples"; }
     private void __BuildControlTree(Control __ctrl) {
           ctrl.SetRenderMethodDelegate(new RenderMethod(this. Render control1));
     private void Render control1(HtmlTextWriter output, Control parameterContainer) {
           output.Write("\r\n<html>\r\n\t<head> <title>Page counter</title> </head>\r\n\t<body>\r\n\t(t) +
           "<h1>Welcome</h1>\r\n\t\tYou are visitor number "):
           output.Write(CounterValue());
           _output.Write(" !\r\n\t</body>\r\n</html>\r\n");
     protected override void FrameworkInitialize() {
           BuildControlTree(this);
        this.FileDependencies = __fileDependencies;
        this.EnableViewStateMac = true; this.Request.ValidateInput();
     int CounterValue() {
         ....
```

School of Engineering

Hintergrund-Code (Code Behind)



```
Counter.aspx
<%@ Page Language="C#" Inherits=" CounterPage" CodeFile="CounterPage.aspx.cs" %>
<html>
   <head> <title>Page counter</title> </head>
   <body>
      <h1>Welcome</h1>
      You are visitor number <%=CounterValue()%>
   </body>
</html>
                                                                    CounterPage.aspx.cs
using System.IO;
public partial class _CounterPage : System.Web.UI.Page {
   public int CounterValue() {
      FileStream s = new FileStream("c:\\Data\\Counter.dat",
                                        FileMode.OpenOrCreate);
      . . .
      n = r.ReadInt32();
      n++;
      . . .
      return n;
```

School of Engineering

Erzeugte Seitenklasse





School of Engineering



Web-Formulare

School of Engineering

© K. Rege, ZHAW

18 von 75

HTML-Formulare



CGI-Programm myprog

liest total und amount

schickt neuen HTML-Text zurück, in dem total und amount neue Werte haben

Probleme

- CGI-Programmierung kompliziert
- auf HTML-Elemente eingeschränkt
- muss Zustand der Textfelder beim Zurückschicken neu aufbauen

School of Engineering

🖉 Kassa - Microsoft Internet Explorer				×		
<u> </u>	<u>E</u> dit	<u>V</u> iew	F <u>a</u> vorites	<u>T</u> ools	<u>H</u> elp	
Kassenstand: 0 Euro						
😂 Done	9			🗏 Local ir	ntranet	_//

Web-Formulare unter ASP.NET



Adder.aspx



Web-Formulare unter ASP.NET



```
Adder.aspx
<%@ Page Language="C#" Inherits=" AdderPage" CodeFile="Adder.aspx.cs"%>
<html>
   <head><title>Kassenstand</title></head>
   <body>
      <form method="post" Runat="server">
          <b>Kassenstand:</b>
          <asp:Label ID="total" Text="0" Runat="server"/> Euro<br><br>
          <asp:TextBox ID="amount" Runat="server"/>
          <asp:Button ID="ok" Text="Enter" OnClick="ButtonClick" Runat="server" />
      </form>
   </body>
</html>
                                                                         Adder.aspx.cs
using System; using System.Web.UI; using System.Web.UI.WebControls;
public particial class _AdderPage : Page {
   public void ButtonClick (object sender, EventArgs e) {
                                                                  🖉 Account - Microsoft Internet Exp... 💶 🗖 🗙
      int totalVal = Convert.ToInt32(total.Text);
                                                                    File Edit View Favorites Tools »
      int amountVal = Convert.ToInt32(amount.Text);
      total.Text = (totalVal + amountVal).ToString();
                                                                   Balance: 100 Euro
}
                                                                                       Enter
                                                                   100
                                                                  🞒 Doni
                                                                                🔠 Local intranet
```

School of Engineering

Zurückgesandtes HTML



Counter.aspx

```
<%@ Page Language="C#"
  Inherits=" AdderPage"
  Src="Adder.aspx.cs"%>
<html>
  <head><title>Account</title></head>
  <body>
    <form method="post" Runat="server">
      <b>Balance:</b>
      <asp:Label ID="total" Text="0"
        Runat="server"/> Euro<br><br>
      <asp:TextBox ID="amount"
        Runat="server"/>
      <asp:Button ID="ok"
        Text="Enter"
        OnClick="ButtonClick"
        Runat="server" />
    </form>
  </body>
</html>
```

Zurückgesandtes HTML

```
<html>
  <head> <title>Account</title> </head>
  <body>
   <form name="_ctl0" method="post"</pre>
      action="Adder.aspx" id="_ctl0">
      <input type="hidden" name=" VIEWSTATE"</pre>
        value="dDwxNTq0NTEzNzMyO30802w8aTwxP" +
        "js+02w8dDw7bDxpPDE+0z47bDx0PHA8cDxs"+
        "PFRleHO7PjtsPDEwMDs+Pjs+Ozs+Oz4+Oz4+" +
        "Oz7uOqbDI3uKWY/X5D1Fw8zmjTZkwq==" />
      <b>Balance:</b>
      <span id="total">100</span>
      <input type="text" name="amount"
        value="100" id="amount" />
      <input type="submit" name="ok"
        value="Enter" id="ok" />
    </form>
 </body>
</html>
```

Schreibweise von Web-Controls



<asp:ClassName PropertyName="value" ... Runat="server" />

Beispiel

<asp:Label ID="total" Text="Hello" ForeColor="Red" Runat="server" />

Klassen im Namensraum System.Web.UI

```
public class Label: WebControl {
  public virtual string ID { get {...} set {...} }
  public virtual string Text { get {...} set {...} }
  public virtual Color ForeColor { get {...} set {...} }
  ...
}
```

Alternative Schreibweise

```
<asp:Label ID="total" ForeColor="Red" Runat="server" >
Hello
</asp:Label>
```

Vorteile von Web-Formularen



Die Seite ist ein Objekt

-> Man kann ihre Properties und Methoden benutzen: *page.User*, *page.FindControl*(), ...

Die GUI-Elemente sind Objekte

-> Man kann ihre Properties und Methoden benutzen: *amount.Text*, *amount.Font*, *amount.Width*, ...

Man kann eigene GUI-Elemente programmieren

Zugriff auf gesamte .NET-Bibliothek möglich Datenbanken, XML, RMI, ...

Zustand der GUI-Elemente bleibt erhalten: amount.Text muss z.B. nicht neu gesetzt werden

Steuerelemente (Web Controls)



Label	abc	Calendar	≤ Febru Mo Di Mi <u>27</u> <u>28</u> <u>29</u>	nar2003 <u>≥</u> Do Fr Sa So <u>30 31 1 2</u>		
TextBox			$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$		
Button						
	Click	DataCrid	EmployeeI	D FirstNam	e LastName	
		DalaGhu	1	Nancy	Davolio	
RadioButton			2	Andrew	Fuller	
RadioBatton			3	Janet	Leverling	
	O Radio		4	Steven	Buchapan	
			6	Michael	Suvama	
CheckBox			7	Robert	King	
			8	Laura	Callahan	
	🗀 Check		9	Anne	Dodsworth	
DropDownList				1		
	Linz 💌	Eigene User Control				
ListRox						
	Äpfel	Eigene Custom Controls				
	Birnen		Bibliothek von Controls http://www.asp.net/Default.asp			
	Zwetschken	Bibliothek vor ///				



Ereignisbehandlung

School of Engineering

© K. Rege, ZHAW

26 von 75

Ereignisgesteuerte Ablauflogik





Ereignisarten



Steuerelement	Ereignis	Wann wird es ausgelöst?
alle	Init Load PreRender Unload	 wenn das Element erzeugt wurde nachdem die vom Browser geschickten Daten in das Element geladen wurden bevor HTML-Code für das Element erzeugt wird bevor das Element aus dem Speicher entfernt wird
Button	Click	wenn der Button geklickt wurde
TextBox	TextChanged	wenn der Inhalt der TextBox verändert wurde
CheckBox	CheckedChanged	wenn sich der Zustand der CheckBox geändert hat
ListBox	SelectedIndexChanged	wenn ein Element aus der Liste ausgewählt wurde

























🚰 Kassenstand - Microsoft Internet Exp 💶 🗙						
<u> </u>	<u>E</u> dit	⊻iew	F <u>a</u> vorites	<u>T</u> ools	<u>H</u> € ≫	-
Kass	sensta	and: 1	00 Euro			_
100				Einza	hlen	-
, 🕘 Done	e		📃 📴 Lo	cal intrar	net	

<html></html>	
 <input type="text"/> <input type="button"/>	



6. Entladen

- löse Unload-Ereignisse für Abschlussarbeiten aus

Server

Client

Welche Ereignisse führen zur Rundreise



Rücksendeereignisse (führen sofort zur Rundreise)

click me 🔨 Click

<asp:Button Text="click me" Runat="server" OnClick="DoClick" />

Verzögerte Ereignisse (werden erst bei der nächsten Rundreise behandelt)



<asp:TextBox Runat="server" OnTextChanged="DoTextChanged" />



SelectedIndexChanged<asp:ListBox Rows="3" Runat="server" OnSelectedIndexChanged="DoSIChanged" />

AutoPostBack (bewirkt, dass verzögertes Ereignis sofort zur Rundreise führt)

abc 🔨 TextChanged

<asp:TextBox Runat="server" **AutoPostBack="true"** OnTextChanged="DoTextChanged" />


Web-Controls

School of Engineering

© K. Rege, ZHAW

37 von 75

Steuerelement-Hierarchie





Klasse Control

```
public class Control: ... {
  public virtual string ID { get; set; }
  public virtual ControlCollection Controls { get; }
  public virtual Control Parent { get; }
  public virtual Page Page { get; set; }
  public virtual bool Visible { get; set; }
  protected virtual StateBag ViewState { get; }
  public virtual bool EnableViewState { get; set; }
  . . .
  public virtual bool HasControls();
  public virtual Control FindControl (string id);
  public virtual void DataBind();
  protected virtual void LoadViewState (object state);
  protected virtual object SaveViewState();
  protected virtual Render (HtmlTextWriter w);
  . . .
  public event EventHandler Init;
  public event EventHandler Load;
  public event EventHandler DataBinding;
  public event EventHandler PreRender;
  public event EventHandler Unload;
   protected virtual void OnInit(EventArgs e);
   protected virtual void OnLoad(EventArgs e);
```



Properties

Name des Steuerelements innere Steuerelemente äusseres Steuerelement Seite, in der das Element vorkommt soll das Element sichtbar sein? Zustandsinformationen über dieses Element soll Zustand in Seitenzustand einfliessen?

Methoden

hat das Element innere Elemente? sucht ein inneres Element mit Namen id übernimmt Daten aus einer Datenquelle lädt Element-Zustand aus Seitenzustand speichert Element-Zustand in Seitenzustand bildet das Element nach HTML ab

Events

nachdem das Element erzeugt wurde nachdem Seitenzustand geladen wurde nachdem DataBind aufgerufen wurde bevor Element nach HTML gerendert wird bevor Element freigegeben wird

On ... Methoden - zum überschreiben nachdem das Element erzeugt wurde nachdem Seitenzustand geladen wurde

Properties der Klasse Control



Containment-Beziehung



ListItem ListItem ListItem

ViewState

```
public void ButtonClick (object Button, EventArgs e) {
    int clicks = ViewState["nClicks"] == null ? 0 : (int) ViewState["nClicks"];
    ViewState["nClicks"] = ++clicks;
}
```

- in *ViewState* lassen sich beliebige Daten ablegen
- ViewState wird in einem versteckten Feld der HTML-Seite gespeichert
- dies hier ist ViewState von Page (ViewState von Button ist leider protected)

Klasse WebControl





WebControl (Fonts)



Fonts

```
public sealed class FontInfo {
   public string Name { get; set; }
   public FontUnit Size { get; set; }
   public bool Bold { get; set; }
   public bool Italic { get; set; }
   public bool Underline { get; set; }
   ...
}
public struct FontUnit {
   public FontUnit (Unit size);
   public FontUnit (FontSize size);
   public Unit Unit { get; }
   public FontSize Type { get; }
   ...
}
public enum FontSize { AsUnit, XSmall,
```

Small, Medium, Large, XLarge, ... }

Setzen in aspx-Datei:

```
<asp:Button ID="b1" Font-Name="Arial"
Font-Size="Large" Font-Bold="true" .../>
<asp:Button ID="b2" Font-Name="Times"
Font-Size="12px" Font-Italic="true" ... />
```

Setzen im Server-Code:

```
bl.Font.Name = "Arial";
bl.Font.Size = new FontUnit(FontSize.Large);
bl.Font.Bold = true;
b2.Font.Name = "Times";
b2.Font.Size = new FontUnit(12);
b2.Font.Italic = true;
```

WebControl (sonstige Properties)



BorderStyle



NotSet	None	Dotted
Dashed	Solid	Double
Groove	Ridge	Inset
Outset		

Enabled

<asp:Button Enabled="false" ... />

enabled disabled

zeigt das Element an, deaktiviert es aber

TabIndex



School of Engineering

Klasse Button







Button (Command-Ereignis)



Command-Ereignis

wenn mehrere Buttons einer Seite denselben Event-Handler aufrufen sollen





```
public void DoCommand (object sender, CommandEventArgs e) {
   double total = Convert.ToDouble(label.Text);
   if (e.CommandName == "add")
      total += total * Convert.ToDouble(e.CommandArgument);
   else if (e.CommandName == "sub")
      total -= total * Convert.ToDouble(e.CommandArgument);
   label.Text = total.ToString("f2");
}
```

School of Engineering

Klasse TextBox





Klasse CheckBox





Klasse RadioButton



<pre>public class RadioButton: CheckBox { public virtual string GroupName { get; set; } }</pre>	Zusammengehörige RadioButtons müssen denselben Gruppennamen haben	



```
if (card.Checked) label.Text += card.Text;
```

Klasse ListControl





DataSource beliebiges Objekt das *ICollection* implementiert (*DataView*, *Array*, *ArrayList*, *SortedList*, ...)

DataTextField bei DataView: Name der Spalte, die anzuzeigenden Text enthält

DataValueFieldbei DataView: Name der Spalte, die den Wert zum angezeigten Text enthält

Klasse ListBox





ListBox (dynamisch spezifizierte Liste)





ListBox (noch einfacher)





ListBox (aus einer Datenbank generierte Liste)



```
<form OnInit="PageInit" Runat="server">
<asp:ListBox ID="list" DataTextField="LastName" DataValueField="EmployeeID"
OnSelectedIndexChanged="HandleSelect" AutoPostBack="true" Runat="server" /><br>
<asp:Label ID="label" Runat="server" />
</form>
```



Klasse DropDownList



public class DropDownList: ListControl {
 // gleiche Schnittstelle wie ListControl
}

Statisch spezifizierte DropDownList

```
<form Runat="server">
<asp:DropDownList ID="list" OnSelectedIndexChanged="HandleSelect"
AutoPostBack="true" Runat="server" >
<asp:ListItem Text="United States" Value="USA" />
<asp:ListItem Text="Great Britain" Value="GB" />
<asp:ListItem Text="Germany" Value="D" />
<asp:ListItem Text="France" Value="F" />
<asp:ListItem Text="Italy" Value="I" />
</asp:DropDownList><br>
<asp:Label ID="lab" Runat="server" />
</form>
```



void HandleSelect (object sender, EventArgs e) {
 lab.Text = "The selected country has the international car code ";
 if (list.SelectedItem != null) lab.Text += list.SelectedItem.Value;
}

DropDownList kann wie ListBox auch dynamisch befüllt werden

School of Engineering

Visualisierung von Datenbank Tabellen



GridView

<asp:GridView DataSourceID="data" Runat="server" AllowSorting="true" />

id	name	subject
9955004	Miller	Computer Science
9934128	Johnson	Economics
9955332	Howard	Computer Science
9882076	Keen	Physics
9834524	Feldman	Economics

- Visualisiert die AccessDataSource namens data
- Erlaubt Sortierung der Spalten durch Klick auf den Spaltennamen
- Kann auf vielfältige Weise formatiert werden

AccessDataSource, SqlDataSource, XmlDataSource



via angezeigt werden kann

School of Engineering

Editieren eines GridViews



<asp:AccessDataSource ID="data" Runat="server" Spaltenwerte
DataFile="db.mdb"
SelectCommand="SELECT id, name, subject FROM Students"
UpdateCommand="UPDATE Students SET name=@name, subject=@subject WHERE id=@id"
DeleteCommand="DELETE FROM Students WHERE id=@id" />
<asp:GridView ID="grid" DataSource="data" Runat="server"
DataKeyNames="id"
AutoGenerateEditButton="true" />

	iđ	name	subject		iđ	name	subject
Edit Delete	9955004	Miller	Computer Science	<u>Edit</u> <u>Delete</u>	9955004	Miller	Computer Science
Edit Delete	9934128	Johnson	Economics	<u>Update</u> <u>Cancel</u>	9934128	Johnson	Economics
<u>Edit</u> Delete	9955332	Howard	Computer Science	<u>Edit</u> <u>Delete</u>	9955332	Howard	Computer Science
<u>Edit</u> Delete	9882076	Keen	Physics	<u>Edit</u> <u>Delete</u>	9882076	Keen	Physics
<u>Edit</u> Delete	9834524	Feldman	Economics	<u>Edit</u> <u>Delete</u>	9834524	Feldman	Economics

Es werden ausserdem Events ausgelöst, die man abfangen und behandeln kann

DropDownList mit AccessDataSource



<asp:AccessDataSource ID="data" Runat="server" DataFile="db.mdb" SelectCommand="SELECT DISTINCT subject FROM Students" />

<asp:DropDownList DataSourceID="data" DataTextField="subject" Runat="server" />

Computer Science 🔽



Validatoren

School of Engineering

© K. Rege, ZHAW

58 von 75

Validatoren



Objekte, die Plausibilitätsprüfungen durchführen



Validatoren (Beispiel)



```
<form runat="server">

Name: <asp:TextBox id="name" runat="server" Width="136px"></asp:TextBox>

<asp:RequiredFieldValidator id="RequiredFieldValidator1" runat="server" ControlToValidate="name"

ErrorMessage="Sie müssen einen Namen eintragen" >

</asp:RequiredFieldValidator>

Alter: <asp:TextBox id="alter" runat="server">0</asp:TextBox>

<asp:RangeValidator id="RangeValidator1" runat="server" ControlToValidate="alter"

MinimumValue="0" MaximumValue="100"

ErrorMessage="Das Alter muss zwischen 0 und 100 liegen" >

</asp:RangeValidator>
```



RequiredFieldValidator & ValidationSummary



Überprüft ob Eingabefeld leer ist

- ControlToValidate="": Steuerelement, das überwacht wird
- ErrorMessage="": Die Fehlermeldung
- display = "none": Fehler wird nicht angezeigt, sondern nur in V.S.
- EnableClientScript = "yes" (default) Es wird JScript für Client erzeugt





RangeValidator und CompareValidator



Überprüft ob Wert in erlaubten Bereich liegt

- gleiche (Grund-)Properties wie RequiredFieldValidator
- Type: String, Integer, Double, Date, Currency
- MinimumValue und MaximumValue
- Achtung: leeres Feld wird nicht geprüft !
- **CompareValidator** vergleicht nur mit einem Wert

Beispiel: Datum in einem Bereich

```
<asp:TextBox ID="date" Runat="server" />
<asp:RangeValidator id="valRange" runat="server" ControlToValidate="date"
ErrorMessage="Das Datum muss zwischen 1.5.2003 und 1.6.2003 liegen"
Type="Date"
MinimumValue="1.5.2003" MaximumValue="1.6.2003" >
</asp:RangeValidator>
```

RegularExpressionValidator



Überprüft ob Wert in erlaubten Bereich liegt

- gleiche (Grund-)Properties wie RequiredFieldValidator
- Eingabe wird mit regulärem Ausdruck verglichen
- Syntax
 - . beliebiger Buchstaben
 - \: Spezialzeichen folgt
 - * : beliebig oft
 - + : mindestens 1-mal
 - ...

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/js56reconRegularExpressions.asp Beispiel: überprüfe ob Feld E-Mail enthält:

```
<asp:TextBox id="email" runat="server"></asp:TextBox>
<asp:RegularExpressionValidator id="emailValidator" runat="server"
ErrorMessage="Keine gültige E-Mail Adresse" ControlToValidate="email"
ValidationExpression=".*@.*\..*">
</asp:RegularExpressionValidator>
```

CustomValidator



Überprüft ob Wert in erlaubten Bereich liegt

- gleiche (Grund-)Properties wie RequiredFieldValidator
- ClientValidationFunction: Name der JavaScript Funktion
- OnServerValidate: Name der C# Methode f
 ür die Validation

Beispiel: überprüfe ob Feld E-Mail enthält (nur @)

```
<asp:textbox id="textbox1" runat="server">
<asp:CustomValidator id="valCustom" runat="server"
ControlToValidate="email"
ClientValidationFunction="ClientValidate"
OnServerValidate="ServerValidate"
ErrorMessage="*This box is not valid" dispaly="dynamic">*
</asp:CustomValidator>
```

```
<html>
<head>
<script language="JavaScript">
function ClientValidate(source,arg) {
arg.lsValid = arg.Value.indexOf("@") >= 0;}
</script>
```

School of Engineering



Arbeiten mit Visual Studio

Visual Studio: Projekt anlegen



Eile	<u>E</u> dit <u>V</u> iew	<u>R</u> efactor	Web <u>s</u> ite	Bu	ild <u>D</u> ebug	<u>T</u> ools	<u>W</u> indow	Community	<u>H</u> elp			
	<u>N</u> ew		+	٦,	Project	C	trl+Shift+N		NET			
	<u>O</u> pen	1	<	3	Web Site	>			9			
	A <u>d</u> d		•	1	<u>F</u> ile		Ctrl+N	e.aspx	1			
	⊆lose				Project Fron	n <u>E</u> xisting	Code	:k(object	: sende			
a	Close Solu <u>t</u> ion									1		
	<u>Save</u> TestPage	e.aspx.cs (Ctrl+S		New Web:	Site		_		_	_	×
	Save TestPage	e.aspx.cs <u>A</u> s	;	rat	Tamalahaa							D D 8-8-
	Ad <u>v</u> anced Sav	e Options		ur i	Lemplates:			• •				22 6-6-
7	Jave H∥	Curro	TILLE T	·	2 Sear	NET Web S Web S mplates) Site ite : Templates.		NSP.	NET Web Service		
					A black AS		leh cite			3		
						/F.1961 99	CD SICC					
					Location:		File System	ו	<u>v</u> 🖸	:\My Documents\WebSite4		Browse
					Language:		Visual C#		v			
											4 _{ок}	Cancel
loc	of Engine	erina										

Visual Studio: GUI mit Drag & Drop zusammenbauen





Visual Studio: HTML-Sicht



			1			
	Start Page Registration.aspx* Registr	ation.aspx.cs*		4 Þ ×	Solution Explorer - Regist	ration 4
	Client Objects & Events	-	(No Events)	I = =		
General 🔺	<%@ Page language	="c#" C	Codebehind="Registration	on.aspx.c	Solution Registratio	n' (1 project)
Pointer	DOCTYPE HTML PU</td <td>BLIC "-</td> <td>-//W3C//DTD HTML 4.0 T</td> <td>ransition</td> <td>E References</td> <td></td>	BLIC "-	-//W3C//DTD HTML 4.0 T	ransition	E References	
, romon	<html></html>				📕 📳 AssemblyInf	ro.cs
					Global.asax	
	<pre></pre>	hForm1 <	(title>		Registration	aspx vsdisco
		O-UCENE	PATORN Contont-"Migro	coft Wign	Web.config	
		- UCODI	KATOR CONCENT- MICLO	SULL VISU		
	<meta nam<="" td=""/> <td>e="CODE</td> <td>LANGUAGE COntent="C</td> <td>#"></td> <td>Properties</td> <td>д</td>	e="CODE	LANGUAGE COntent="C	#">	Properties	д
	<pre><meta nam<="" pre=""/></pre>	e="vs_c	efaultClientScript" c	ontent="J	DOCUMENT	
	<pre><meta nam<="" pre=""/></pre>	e="vs_t	argetSchema" content=	"http://s	P 🛃 🔲 🖻	
					aLink	
	<pre><body ms_posi<="" pre=""></body></pre>	TIONING	="GridLayout">		aspCompat	
	<form id="</td"><td>"Form1"</td><td>' method="post" runat=</td><td>"server"></td><td>background</td><td></td></form>	"Form1"	' method="post" runat=	"server">	background	
	<asp:< td=""><td>Label i</td><td>d="lab1" style="Z-IND</td><td>EX: 101;</td><td>bgColor</td><td></td></asp:<>	Label i	d="lab1" style="Z-IND	EX: 101;	bgColor	
	<asp:< td=""><td>TextBox</td><td><pre>d id="TextBox2" style=</pre></td><td>"Z-INDEX:</td><td>bottomMargin</td><td></td></asp:<>	TextBox	<pre>d id="TextBox2" style=</pre>	"Z-INDEX:	bottomMargin	
	<asp:< th=""><th>TextBox</th><th><pre>x id="name" style="Z-I</pre></th><th>NDEX: 102</th><th>buffer</th><th>True</th></asp:<>	TextBox	<pre>x id="name" style="Z-I</pre>	NDEX: 102	buffer	True
	<asp:< td=""><td>Label i</td><td>d="lab2" style="Z-IND</td><td>EX: 103:</td><td>charset</td><td></td></asp:<>	Label i	d="lab2" style="Z-IND	EX: 103:	charset	
	<asn:< td=""><td>TextBox</td><td>/ id="TextBox1" style=</td><td>"Z-INDEX:</td><td>clientTarget</td><td></td></asn:<>	TextBox	/ id="TextBox1" style=	"Z-INDEX:	clientTarget	
	<pre></pre>	Lahel i	d="lab3" style="Z-IND"	EX: 105:	codeBehind	Registration.a
	(asp.	Button	id="buttop" style="7-	INDEX: 1007		
	<pre></pre>	BULLOII	THE DUCCON SUFTEE 7-	THUEA. TO	Culture	
					debug	False
	<ybod></ybod>			_	defaultClientScript	JScript
				► E	Description	
					dir	

68 von 75

Visual Studio: Ereignisbehandlung



Doppelklick auf Button legt EventHandler im Hintergrundcode an



School of Engineering

Visual Studio: Ausführen



Menü: Debug | Start

🖉 WebForm1 - Microsoft Internet Explorer 📃 🔍
Eile Edit View Favorites Tools Help
📙 🖶 Back 🔻 🤿 🖌 🙆 🖓 Search 🕋 Favorites 🎯 Media 🎯 🛃 🗸 👋
Address 🕘 http://localhost/Registration/Registration.aspx 💌 🔗 Go 🗍 Links 🎽
Name Register
Phone
🙋 Done 🛛 👘 🖉 Local intranet 🥢

Zusammenfassung



- Erlaubt "gewohnte" objektorientierte Programmierung des HTML GUIs
- Es werden Ereignisse erzeugt, die auf dem Server abgefangen werden können
- Trennung zwischen Layout (HTML) und Logik (z.B. C#)
- compilierte statt interpretierte Server-Scripts
- Es existiert eine grosse Auswahl von vorgefertigten Steuerelementen
- Entwicklung mit VS sehr konfortabel

Fragen?





School of Engineering
ASP.NET Page Direktiven



Directive Name	Description
@Page	Compilation and processing options
@Import	Replaces C#'s using statement
@Assembly	Replaces csc.exe's /r: switch
@OutputCache	Controls ASP.NET's output caching
@Register	Registers tag prefix/name to class name



School of Engineering

© K. Rege, ZHAW

@Page Attributes



Name	Description
Language	Programming language to use for <%
Buffer	Response buffering
ContentType	Default Content-Type header (MIME)
EnableSessionState	Turns session state on/off
Transaction	Controls COM+ transaction affiliation
CodeFile	Indicates source file for code-behind
Inherits	Name of alternative base class
ClassName	Name of generated class
ErrorPage	URL for unhandled exception page
AspCompat	Run on classic ASP STA thread (slow)

<%@page language="C#" buffer="true" contenttype="text/xml" %>

<%@page language="C#" CodeFile="test.cs" inherits="MyBasePage" %>

Anlegen eines virtuellen Verzeichnisses für IIS



(unter Windows 2000)



Schritte zum Anlegen eines virtuellen Verzeichnisses

Control Panel

- > Administrative Tools
 - > Computer Management

Rechts-Klick auf *Default Web Site* > New ... Virtual Directory

Durch den Dialog führen lassen

Alle aspx-Dateien müssen in einem virtuellen Verzeichnis stehen

Ansprechbar als http://<site-url>/<virtualDirName>/myfile.aspx

© K. Rege, ZHAW

75 von 75