

Web aplikacije

11/13

Izrada Web aplikacija pomoću ASP.NET-a

❑ Što je Web aplikacija?

- Programska aplikacija kojoj se pristupa preko Web preglednika (browsera) ili drugog programa koji implementira HTTP (Hyper Text Transfer Protocol)

❑ Da li je skup Web stranica ujedno i Web aplikacija?

- Aplikacija koristi programsku logiku da bi prikazala sadržaj korisniku
- Kod Web aplikacija obično se izvršava neki programski kod na serveru
- Primjer: statički cjenik nije aplikacija; dinamički cjenik je aplikacija

❑ Primjer Web aplikacije: <http://ahyco.fer.hr>, s funkcijama

- Identifikacija (autorizacija) korisnika
- Pisanje provjere, pregled rezultata, ...
- Baza podataka + programska logika za generiranje Web stranica

❑ Elementi web stranice

- HTML (HyperText Markup Language): osnovni jezik za definiranje Web stranica
- JavaScript: jezik za klijentske skripte koje izvodi preglednik
- C# kôd (ili kôd u nekom drugom jeziku) koji se izvodi na poslužitelju

Karakteristike Web aplikacija

❑ Web aplikacija

- Izvršava se na poslužitelju i na klijentu
- Korisnik može biti bilo tko s pristupom Internetu
- Klijentski dio mora biti napisan u nekom od jezika koji Web preglednik podržava (HTML, JavaScript).
- Ograničen pristup resursima na strani klijenta (npr. JavaScript koji se izvršava unutar preglednika ne može čitati s korisnikovog diska)

❑ Prednosti Web aplikacija

- Jednostavno instaliranje, održavanje i nadogradnja na novu verziju
- Nema instalacijske procedure na strani klijenta (instalira se preglednik)
- Koriste se na bilo kojem OS, koji ima Internet preglednik

❑ Nedostatci Web aplikacija

- Složenija izrada u odnosu na samostojne klijentske aplikacije
- Sigurnosni problemi (neovlašten pristup aplikaciji, računalu-poslužitelju, zatrpavanje prometom)
- Potreba za posebno dizajniranim sučeljem (Web design)
- Mogući problemi pri prikazu u starijim preglednicima
- Potrebna prilagodba regionalnim posebnostima korisnika

ASP.NET

❑ ASP.NET - Active Server Pages .NET

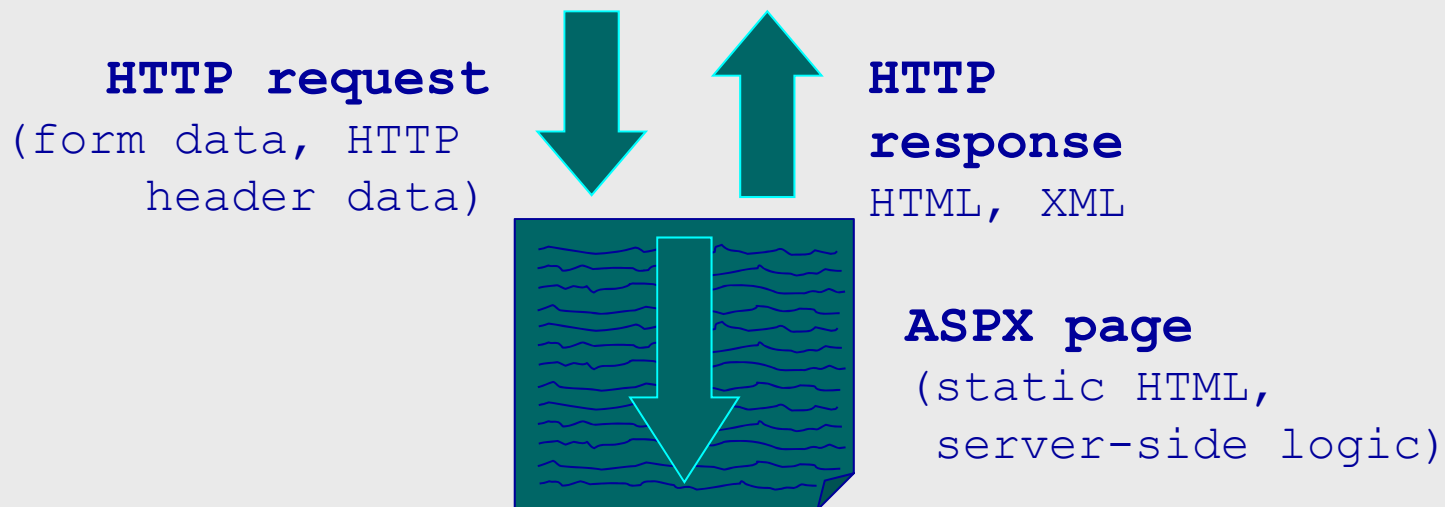
- dinamičko generiranje stranica na poslužitelju
- povrh .NET Frameworka i IIS poslužitelja (Internet Information Services)

❑ Elementni aplikacija: Web Forms i Web controls

- poslužiteljska logika se piše u nekom od .NET jezika (C#, ...)
 - prevođenje → brže izvođenje nego kod skriptnih jezika (ASP, PHP)
- oslonac na ADO.NET – povezivanje s bazama podataka

❑ Podrška za XML Web servise

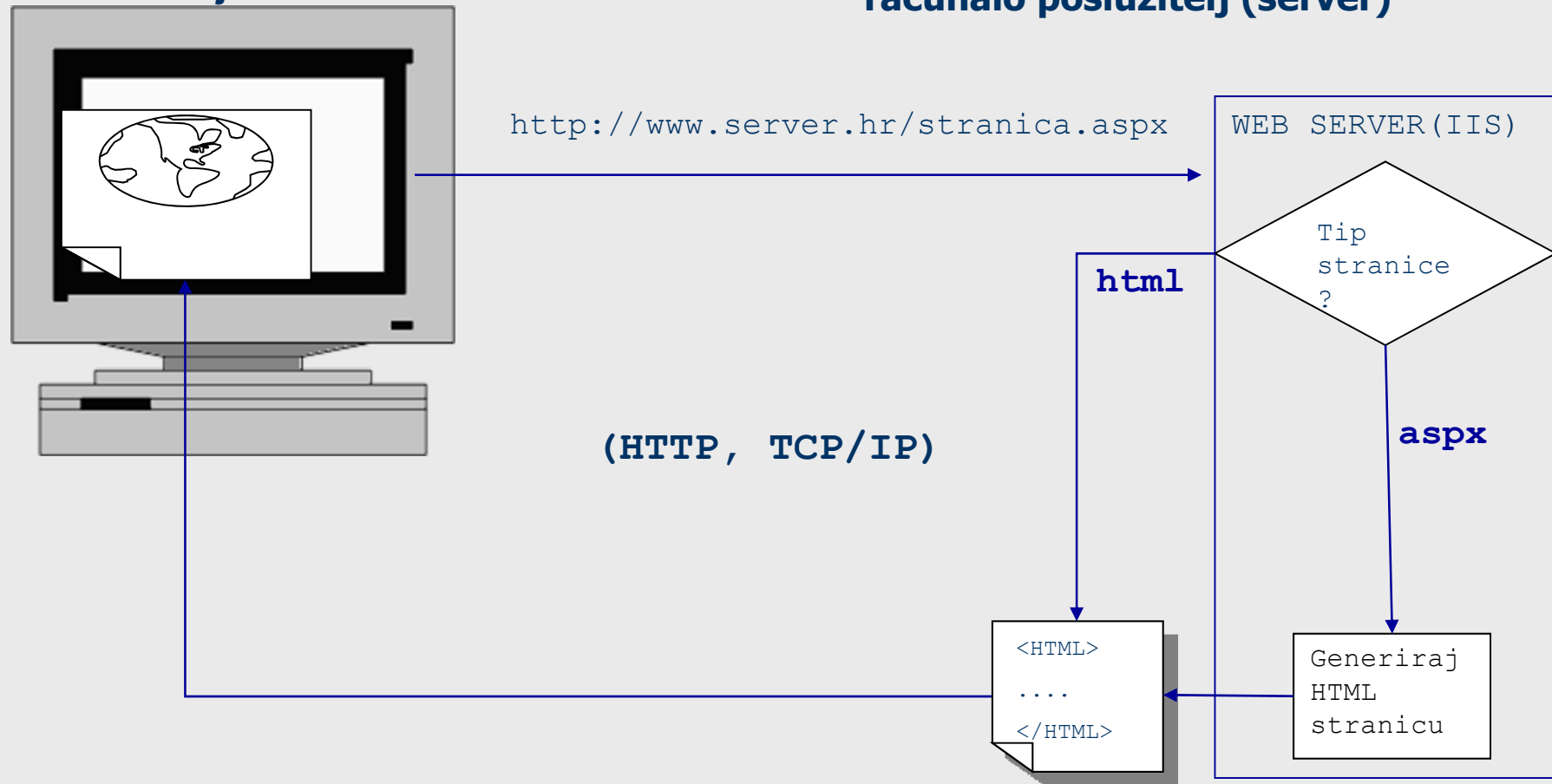
❑ Podrška za mobilne uređaje (WAP, PDA)



Način rada Web aplikacije

računalo klijent

računalo poslužitelj (server)



Internet Information Services (IIS) Server

❑ Administriranje - Microsoft Management Console Snap-In

- My Computer / Manage ili Control Panel / Administrative Tools

❑ Organizacija stranica

- Virtualna kazala – mapiranje URL-a i puta do datoteka
 - URL – Unified Resource Locator
- npr. <http://www.zpr.fer.hr/> nalazi se u kazalu c:\inetpub\wwwroot poslužitelja

❑ Pristup stranicama

- <http://localhost/NazivAplikacije>
 - localhost - Web poslužitelj na lokalnom računalu
- <http://URLposlužitelja/NazivAplikacije>
 - URL poslužitelja – logička adresa poslužitelja u mreži

❑ Pregled stranica

- poslužitelj pregledniku dostavlja HTML (+CSS +JavaScript)

Web Application ili Web Site?

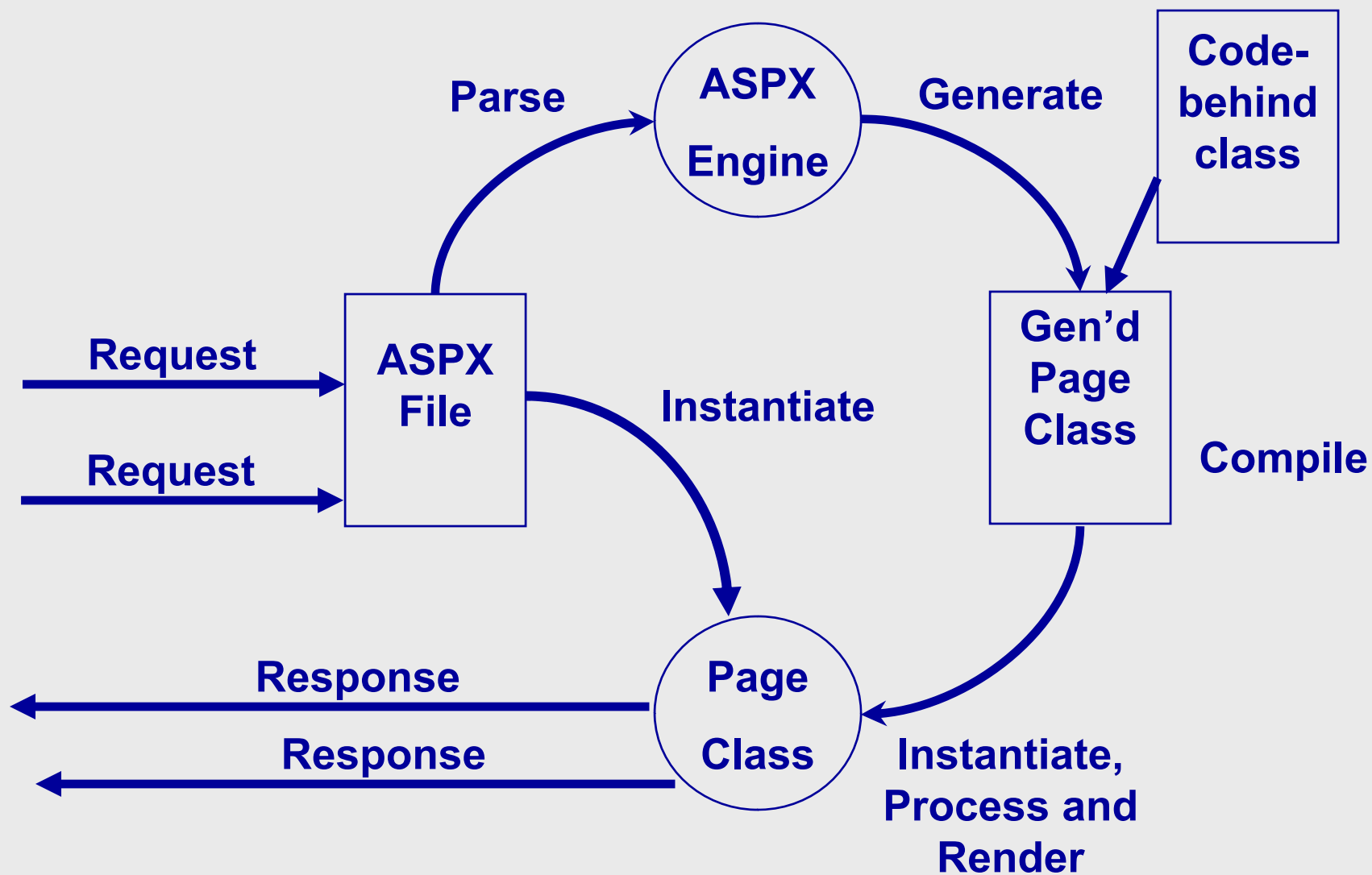
❑ New \ Project \ Web \ ASP.NET Web Application

- Stranica.aspx + Stranica.aspx.cs + Stranica.aspx.designer.cs (CodeBehind)
 - Automatsko dodavanje prostora imena pri kreiranju pojedine stranice/kontrole
 - Naziv razreda oblika WebAppName.Putanja.Stranica
- Eksplicitno dodavanje datoteka u projekt
- Jedan dll nakon Builda
- Kompilacija čitave aplikacije
- Mogućnost definiranja događa prije i poslije izgradnje aplikacije (PreBuild & PostBuild)

❑ New \ WebSite... \ ASP.NET Web Site

- Nije potreban project/solution
- Stranica.aspx + Stranica.aspx.cs (CodeFile)
 - Naziv razreda oblika Putanja_Stranica
- Sve datoteke unutar Web mjesta su automatski dio “projekta”
- Datoteke s razredima nevezanim za pojedinu stranicu smještaju se u mapi App_Code
- “In Place” kompilacija
- App_Code.dll + po jedan dll za svaku stranicu

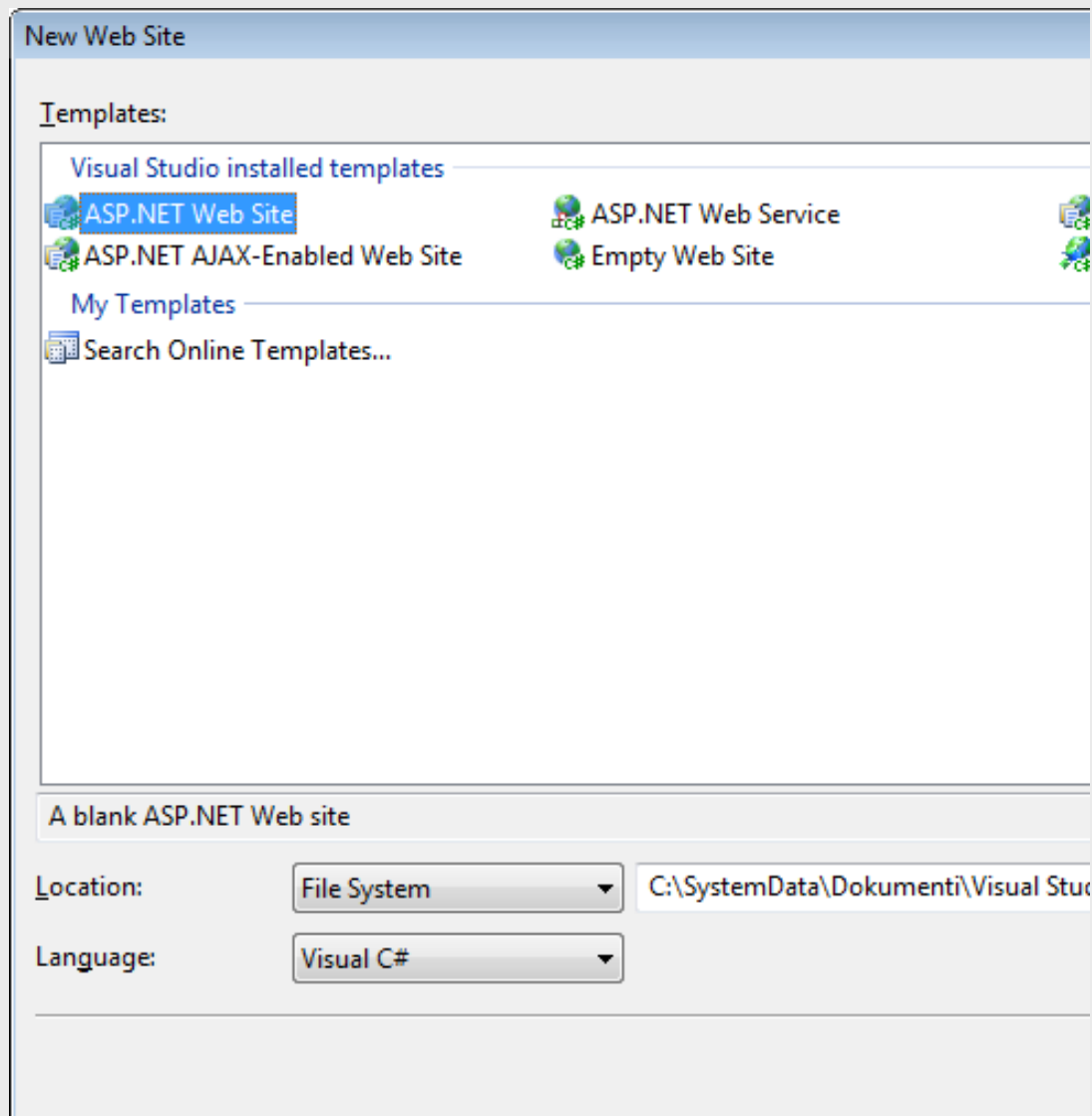
Prevođenje i prikaz stranica



Izrada Web aplikacije

❑ New \ Web site ...

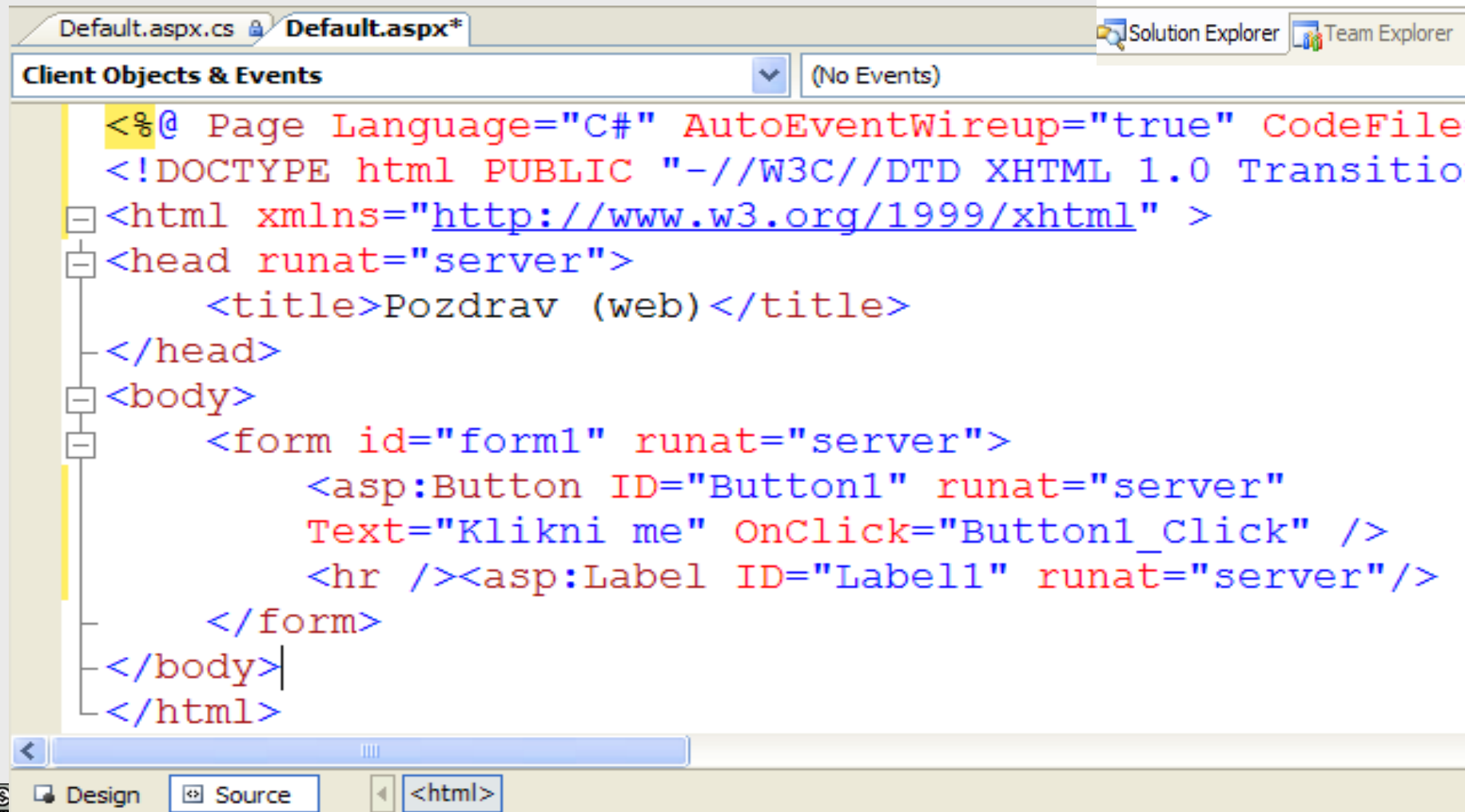
- odabir jezika (C#, VB.NET)
- odabir lokacije:
 - Datotečni sustav – koristi ASP.NET Development Server (privremeni web server) za izvršavanje
 - Smještaj na lokalnom ili udaljenom Web serveru (IIS Server)



Odvajanje dizajna od koda

□ Primjer: Web\PozdravWeb - Default.aspx

- ASPX stranica uobičajeno sadrži HTML, serverske kontrole, JavaScript i ostale prezentacijske elemente



```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transiti
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Pozdrav (web)</title>
</head>
<body>
    <form id="form1" runat="server">
        <asp:Button ID="Button1" runat="server"
            Text="Klikni me" OnClick="Button1_Click" />
        <hr /><asp:Label ID="Label1" runat="server"/>
    </form>
</body>
</html>
```

Odvajanje dizajna od koda

☐ Datoteka s kôdom (CodeFile, Code behind): Default.aspx.cs

- sadrži kôd koji se izvršava na serveru

☐ Parcijalni razredi

- Automatski generirani dio za deklaracije serverskih kontrola i njihovo povezivanje s odgovarajućim događajima je skriven te se naknadno povezuje s kôdom napisanim u parcijalnom razredu.

```
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.Text = "Pozdrav, trenutno vrijeme je " +
                      DateTime.Now.ToString();
    }
}
```

Odvajanje dizajna od koda

❑ Single-File Page model – spoj koda i prezentacije jedne stranice

- Prednost je za male stranice, ali se ne preporuča

```
<%@ Page Language="C#" AutoEventWireup="true" %>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server"><title>Pozdrav (web)</title></head>
<body>
    <script runat="server">
        protected void Button1_Click(object sender, EventArgs e) {
            Label1.Text = "Pozdrav, trenutno vrijeme je " +
                DateTime.Now.ToString();
        }
    </script>
    <form id="form1" runat="server">
        <asp:Button ID="Button1" runat="server" Text="Klikni me"
            OnClick="Button1_Click" />

        <hr />
        <asp:Label ID="Label1" runat="server"/>
    </form>
</body></html>
```



Generirani HTML pri otvaranju stranice

❑ Primjer: (Internet Explorer - View / Source)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head><title>
    Pozdrav (web)
</title></head>
<body>
    <form name="form1" method="post" action="Default.aspx"
id="form1">
<div>
<input type="hidden" name="VIEWSTATE" id="VIEWSTATE"
value="/wEPDwUKLTMyNDcxODYzNWRkStCZGd6Msi68F++I1HdDVcPifBg=" />
</div>
        <input type="submit" name="Button1" value="Klikni me"
id="Button1" />
        <hr />
        <span id="Label1"></span>

<div>
        <input type="hidden" name="__EVENTVALIDATION"
id="__EVENTVALIDATION"
value="/wEWAglAn9rtAQKM54rGBughwZwltKW5yEelZ9bwGuOz2H+L" />
</div></form>
</body>
</html>
```

Generirani HTML pri povratku s poslužitelja

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head><title>
    Pozdrav (web)
</title></head>
<body>
    <form name="form1" method="post" action="Default.aspx"
id="form1">
<div>
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"
value="/wEPDwUKLTMyNDcxODYzNQ9kFgICAw9kFgICAw8PFgIeBFRleHQFI0hlbGxvL
CBpdCBpcyBub3cgMjIuMS4yMDA4IDIwOjQ3OjAzZGRkaSMAh4hEGLWZkJB/ZQ8mQbE3y
kU=" />
</div>
        <input type="submit" name="Button1" value="Klikni me"
id="Button1" />
        <hr />
        <span id="Label1">Pozdrav, trenutno vrijeme je 22.1.2008
20:47:03</span>
</div>
        <input type="hidden" name="__EVENTVALIDATION"
id="__EVENTVALIDATION"
value="/wEWAqKJxK28DAKM54rGBrcYckOj0FPYeuHKSH5w+dcpsMaq" />
</div></form>
</body>
</html>
```

Sintaksa stranice

❑ Osnova stranice je HTML tekst

- Bilo koja HTML stranica može biti prozvana .aspx

❑ Standardne HTML oznake: <oznaka> [</oznaka>]

- <html> - HTML dokument
- <head> - zaglavlje HTML dokumenta
- <title> - naslov HTML dokumenta
-
 - skok u novi redak
- <p> - paragraf, odjeljak teksta
- - generički spremnik

❑ Stranica sadrži specijalne ASP.NET oznake

- Direktive (Directives): <%@ Page Language="C#" %>
- Serverske kontrole: <asp:Button runat="server">
- Odsječke kôda: <script runat="server">...</script>
- Izraze za povezivanje podataka: <%# %>
- Komentare kôda koji se izvršava na serveru: <%-- --%>
- "Mješoviti" kôd (isprepleteni html i .NET kod) <%= %> i <% %>
 - naslijeđeno iz ASP i treba izbjegavati
 - koristiti kôd u zasebnoj datoteci ili <script runat="server"> i kod za obradu događaja

Neka svojstva stranice

❑ `<%@ Page attribute="value" [attribute="value"...] %>`

- `AspCompat` - kompatibilnost s ASP, "true" izaziva pad performanci
- `AutoEventWireup` – automatsko povezivanje kontrola s događajima
- `Buffer` - korištenje međuspremnik za HTTP odzive
- `CodeFile` – izvorna datoteka s pozadinskim kodom
 - pr. `CodeFile="Default.aspx.cs"`
- `CodePage` – kodna stranica
- `EnableViewState` – automatizirano održavanje sadržaja pri opetovanom zahtjevu za stranicom, standardno "true"
- `ErrorPage` – URL na koji se preusmjerava u slučaju neobrađene pogreške
- `Inherits` – osnovni razred Page objekta
- `Language` – programski jezik
 - pr. `<%@ Page language="c#"`
- `MasterPageFile` – putanja do predloška s izgledom
- `Trace` – aktivira praćenje izvođenja stranice

❑ Smije postojati samo jedna *Page* direktiva u `.aspx` datoteci

Web Forms

❑ Web Forms - forme za izradu Web aplikacija

- Web stranica je objekt, izveden iz `System.Web.UI.Page`

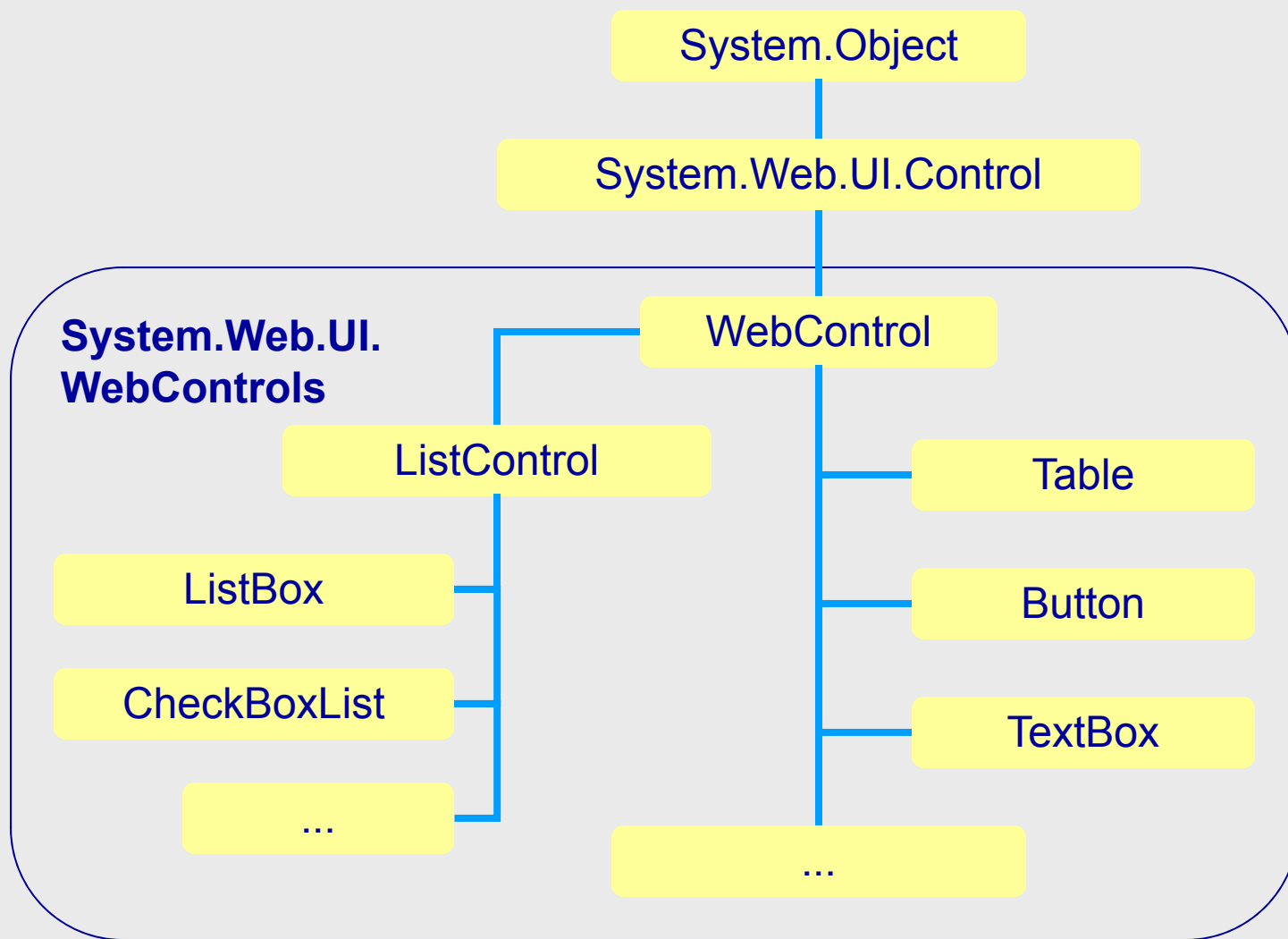
❑ Razvoj stranica korištenjem kontrola, slično Windows Forms

- kontrole su objekti, izvedeni iz `System.Web.UI.Control`
- za svaku kontrolu postoji zasebni razred
 - konzistentniji objektni model, bolja provjera tipova pri prevođenju
- Obrada događaja standardno se obavlja na poslužitelju

❑ Realizacija

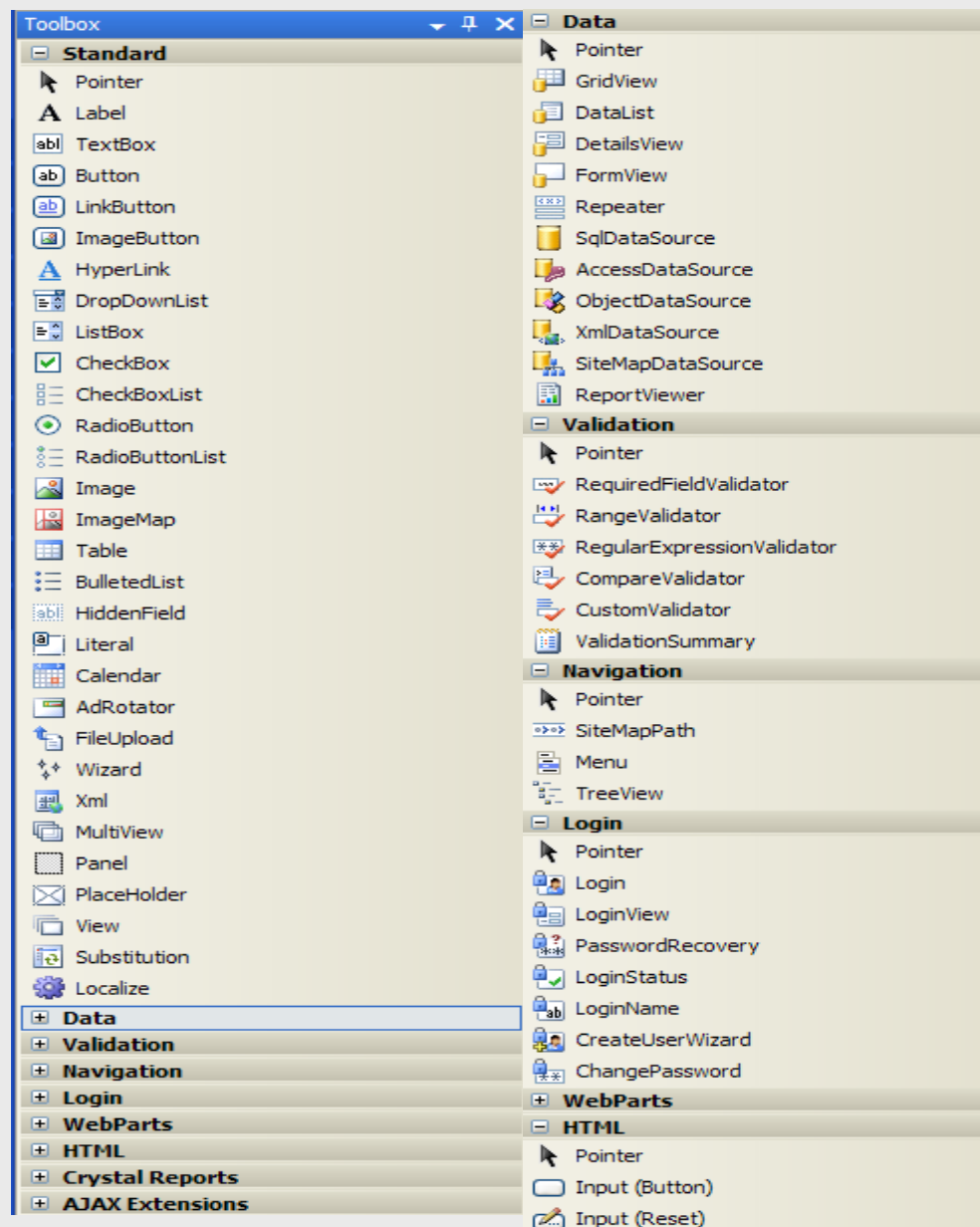
- .NET Framework vrši transformaciju kontrola u HTML
- kontrole imaju mogućnosti HTML kontrola uz dodatnu funkcionalnost
 - npr. `AutoPostBack`, validacijske kontrole, dodatni postupci
- `PostBack` - promjene na stranici šalju se poslužitelju koji obrađenu stranicu vraća pregledniku
- Da bi se očuvao sadržaj stranice, podaci se prenose zajedno sa stranicom
- Za rekonstrukciju sadržaja koriste se skrivena polja (`__VIEWSTATE`)

Hijerarhija razreda



Serverske kontrole

- ❑ ASP.NET sadrži šezdesetak unaprijed definiranih serverskih kontrola podijeljenih u nekoliko grupa



Sintaksa kontrola

❑ Kontrole se definišu kao XML oznake, slično kao HTML oznake (tags)

- `<Kontrola id="naziv" runat="server" [attribute(s)]></Kontrola>`
`<asp:TextBox id="Textbox1" runat="server" />`
- `<Kontrola id="naziv" runat="server" [attribute(s)]/>`
`<asp:Button id="Button1" runat="server" Text="Pošalji"></asp:Button>`

❑ Kontrole su implementirane ASP.NET razredima

- za imenovanje kontrole koristi se atribut `id` uz oznaku da se kontrola generira na serveru `runat="server"`
- oznake i atributi razlikuju velika i mala slova

❑ Svojstva kontrola mogu se postavljati programski

```
txtKomada.Text = "0";  
txtCijena.Text = "100";
```

❑ Obrada događaja obavlja se pozadinskim kodom

```
<asp:Button ID="Button1" runat="server"  
OnClick="Button1_Click" />  
  
...  
private void Button1_Click(object sender, EventArgs e) {  
    labRez.Text = (Convert.ToInt32(txtKomada.Text)  
        * Convert.ToDouble(txtCijena.Text)).ToString();  
}
```

Standardni atributi web kontrola

☐ Proširenje svojstvima za kontrolu prikaza i formata

- Font, BackColor, ForeColor
- BorderColor, BorderStyle, BorderWidth
- Style, CssClass
- Height, Width
- Visible, Enabled

Osnovne Web kontrole

❑ **System.Web.UI.WebControls**

❑ **Web kontrole za unos podataka s forme:**

- TextBox (generirat će HTML nalik na `<INPUT TYPE="text">`)
- Button (`<INPUT TYPE="submit">`)
- CheckBox (`<INPUT TYPE="checkbox">`)
- DropDownList (`<SELECT>...</SELECT>`)
- ...

❑ **Web kontrole koje omogućavaju povezivanje s izvorom podataka:**

- GridView
- DataList
- Repeater
- FormView
- ...

❑ **Ostale Web kontrole:**

- Label (običan tekst)
- HyperLink (` VidljiviTekst `)
- ...

Glavne kategorije kontrola

❑ Intrinsic controls

- Kontrole koje odgovaraju postojećim HTML kontrolama uz određeno povećanje funkcionalnosti (`Label`, `TextBox`, `Panel`, `Image`)

❑ List controls

- Kontrole za rad s različitim listama podataka

❑ Rich controls

- Složenije kontrole, koje osim HTML kontrola automatski generiraju i određeni JavaScript kôd (`Calendar`, `AdRotator`)

❑ Validation controls

- Kontrole za provjeru valjanosti unosa podataka na formi prije nego što se izvrši postback. Uz nekoliko gotovih, moguće napisati i vlastite validacijske kontrole

Intrinsic Controls

❑ Odgovaraju HTML kontrolama

- `<asp:Button>`
- `<asp:ImageButton>`
- `<asp:LinkButton>`
- `<asp:HyperLink>`
- `<asp:TextBox>`
- `<asp:CheckBox>`
- `<asp:RadioButton>`
- `<asp:Image>`
- `<asp:Label>`
- `<asp:Panel>`
- `<asp:Table>`

❑ `TextBox`, `ListControl`, `CheckBox` standardno nemaju postavljen automatski *postback*

- ručno postavljanje svojstva `AutoPostBack=true` za aktiviranje događaja

List Controls

❑ Liste podataka

- `<asp:DropDownList>`
- `<asp:ListBox>`
- `<asp:RadioButtonList>`
- `<asp:CheckBoxList>`
- `<asp:Repeater>`
- `<asp:DataList>`
- `<asp:GridView>`

❑ `CheckBoxList` i `RadioButtonList`

- kolekcije koje sadrže elemente tipa `CheckBox`, odnosno `RadioButton`
- mogu se povezati na podatke (data binding)

❑ `Repeater`, `DataList` i `GridView`

- snažne, korisnički prilagodljive kontrole za povezivanje podataka i njihov prikaz u proizvoljnom obliku (`Repeater`), u obliku tablice (`DataList`) ili u obliku mreže (`GridView`)
- prilagodba se može obaviti predlošcima
- mogu sadržavati druge kontrole

Rich Controls

❑ Korisničke (custom) kontrole

- `<asp:calendar>` - slično `MonthCalendar` u Windows Forms
- `<asp:adrotator>` - automatsko prikazivanje oglasa definiranih u XML datoteci čije je ime navedeno u svojstvu kontrole `AdvertisementFile`

```
<Advertisements>
```

```
<Ad>
```

```
<ImageUrl>../images/banner.gif</ImageUrl>
```

```
<NavigateUrl>http://www.microsoft.com/NET</NavigateUrl>
```

```
<AlternateText>Microsoft .NET Site</AlternateText>
```

```
<Impressions>80</Impressions>
```

```
<Keyword>Microsoft.NET</Keyword>
```

```
<Caption>This is a caption for Microsoft.NET</Caption>
```

```
</Ad>
```

```
...
```

```
</Advertisements>
```

- ostale nadodane (3rd party) kontrole

Validacijske kontrole

❑ Provjera sadržaja drugih kontrola

- kontrole izvedene iz `System.Web.UI.WebControls.BaseValidator`
- validacija se deklarira izvan kontrole za unos podataka

❑ `<asp:RequiredFieldValidator>`

- provjera obaveznog unosa

❑ `<asp:RangeValidator>`

- provjera na dozvoljeni raspon vrijednosti
- svojstva `MinimumValue`, `MaximumValue`

❑ `<asp:CompareValidator>`

- usporedba s konstantom, drugom kontrolom ili tipom podatka
- svojstva `Operator`, `Type` i `ValueToCompare`
- `Operator` – `Equal`, `NotEqual`, `GreaterThan` itd.
- `Type` – `Currency`, `Date`, `Double`, `Integer`, `String`

Validacijske kontrole (nastavak)

❑ `<asp:RegularExpressionValidator>`

- provjera da li vrijednost poštuje uzorak (pattern)
- svojstvo `ValidationExpression`, npr. `\d{5}(-\d{4})?`

❑ `<asp:CustomValidator>`

- korisnički definirana validacijska funkcija

❑ `<asp:ValidationSummary>`

- jedinstvena lista poruka o validacijskim pogreškama
- `DisplayMode: BulletList, List, SingleParagraph`

Primjer validacijskih kontrola

❑ Ključna svojstva validacijskih kontrola

- `Text` – tekst koji se prikazuje u trenutku pojedinačne provjere
- `ErrorMessage` – tekst za `ValidationSummary`
- `ControlToValidate` – naziv kontrole koju treba provjeriti
- `Display` – način prikaza vrijednosti svojstva `Text`
 - `Static`: tekst uvijek na istom mjestu
 - `Dynamic`: izgled stranice (širina i prijelom) se mijenja ovisno o tekstu
 - `None`: ne prikazuje se `Text`, ali se može koristiti `ValidationSummary`

❑ Za istu kontrolu može se koristiti više validatora

❑ Stranica ima

- Postupak `Page.Validate()` ; - obavlja validaciju
- Svojstvo `Page` je instanca stranice koja sadrži serversku kontrolu
- Svojstvo `Page.IsValid` – oznaka da su sve provjere uspješne

❑ Kontrole za napuštanje stranice mogu imati postavljeno svojstvo `CausesValidation=True`

Primjer validacijskih kontrola

- ❑ Ugrađene validacijske kontrole se izvršavaju na klijentu unutar web preglednika (automatski generiran odgovarajući JavaScript kôd)
- ❑ Moguće je definirati vlastite validacijske kontrole (izvršavaju se na serverskoj strani – nakon *postbacka*)
- ❑ **Primjer:**

```
private void CustomValidator1_ServerValidate
(object source, ...ServerValidateEventArgs args) {
    int num = Int32.Parse(args.Value);
    if ((num % 2) == 0)
        args.IsValid = true;
    else
        args.IsValid = false;
}

private void Page_Load(object sender, System.EventArgs e) {
    if (Page.IsPostBack) {
        Page.Validate();
        Msg.Text = "Page.IsValid = " + Page.IsValid.ToString();
    }
}
```

Postupak izrade web stranica

1. Izbor željenih kontrola (Label, DropDownList, GridView)
2. Imenovanje kontrola koje referenciramo u kôdu
3. Dizajn forme - razmještaj kontrola i postavljanje svojstava
4. Programiranje događaja na poslužitelju

☐ Redoslijed događaja (**poslužitelj**, **mreža**, **klijent**):

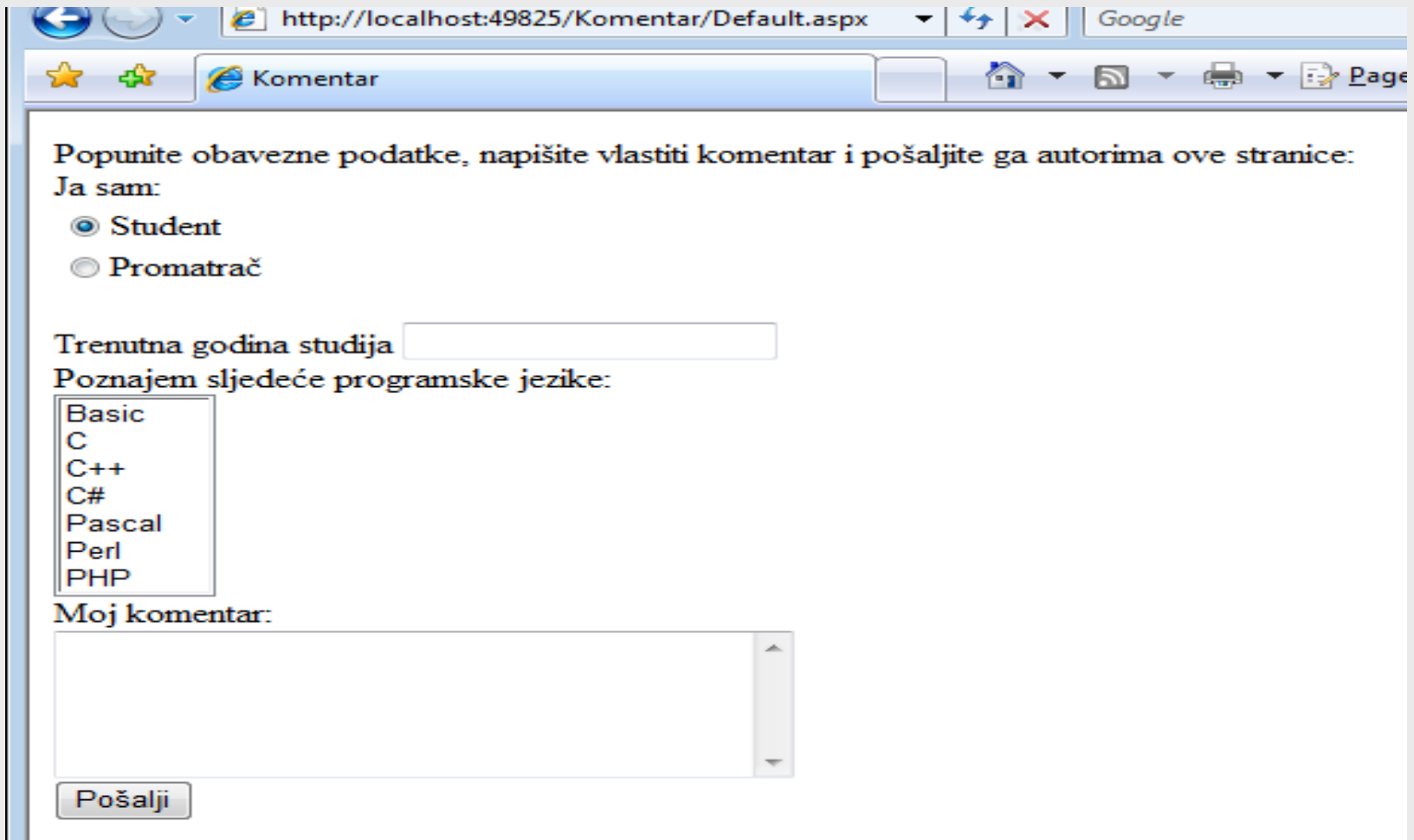
1. **zahtjev za stranicom se šalje poslužitelju**
2. **Page_Load (IsPostBack je False)**
3. **stranica se šalje klijentu**
4. prikaz stranice u pregledniku
5. događaj promjene (npr. Textbox1_Changed) ili akcija (Button1_Click) aktivira slanje
6. **zahtjev za stranicom se šalje poslužitelju (zajedno s podacima)**
7. **Page_Load (IsPostBack je True)**
8. **obrada događaja kontrole (npr. Button1_Click)**
9. **stranica se šalje klijentu**
10. prikaz stranice u pregledniku

Primjer korištenja serverskih kontrola

❑ Primjer:  Web\Komentar – Default.aspx

❑ Korištene kontrole:

- RadioButtonList, ListBox, Button, Label
- RequiredFieldValidator, RegularExpressionValidator



Popunite obavezne podatke, napišite vlastiti komentar i pošaljite ga autorima ove stranice:

Ja sam:

☒ Student

☐ Promatrač

Trenutna godina studija

Poznajem sljedeće programske jezike:

Basic
C
C++
C#
Pascal
Perl
PHP

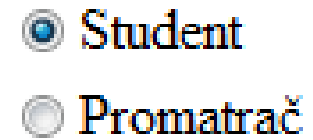
Moj komentar:

Pošalji

RadioButtonList

- ❑ Služi za odabir samo jedne od ponuđenih opcija. Svaki `Listitem` predstavlja jednu od opcija.

```
<asp:RadioButtonList ID="rblStatus" runat="server">  
  <asp:ListItem Value="S">Student</asp:ListItem>  
  <asp:ListItem Value="P">Promatrač</asp:ListItem>  
</asp:RadioButtonList>
```



- ❑ *ListItem* može imati sljedeća svojstva:

- `Selected` (True/False) : Da li je opcija inicijalno označena? Ako više opcija ima postavljenu vrijednost `Selected` na `True`, bit će označena zadnje navedena.
- `Enabled` (True/False) : Da li je moguće odabrati navedenu opciju? Ako nije, navedena opcija se pojavljuje zasivljena.
- `Value` : string koji predstavlja vrijednost navedene opcije
- `Text`: tekst koji se prikazuje pored navedene opcije. U slučaju da je kod opcije između otvorenog i zatvorenog taga naveden neki tekst, svojstvo `Text` poprima tu vrijednost

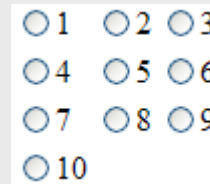
RadioButtonList

❑ Važnija svojstva kontrole RadioButtonList su:

- RepeatDirection (Vertical/Horizontal): redoslijed prikaza elemenata
- RepeatColumns: ograničenje na broj stupaca (pretpostavljena vrijednost je 0 koja označava da nema ograničenja)

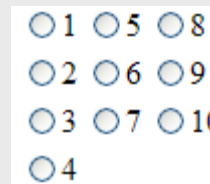
`RepeatDirection="Horizontal"`

`RepeatColumns="3"`



`RepeatDirection="Vertical"`

`RepeatColumns="3"`



- AutoPostBack: promjena označenog elementa uzrokuje trenutni postback

❑ Događaj koji evidentira promjenu označene opcije:

`SelectedIndexChanged`

Primjer kontrole *RadioButtonList*

- ❑ Klikom na *RadioButtonList* s opcijama Student/Promatrač automatski će se dogoditi *postback* te izvršiti postupak koji će pokazati/sakriti tekstualno polje za unos godine studija

```
<asp:RadioButtonList ID="rblStatus" runat="server"
    AutoPostBack="True"
    OnSelectedIndexChanged="rblStatus_SelectedIndexChanged">
    <asp:ListItem Value="S">Student</asp:ListItem>
    <asp:ListItem Value="P">Promatrač</asp:ListItem>
</asp:RadioButtonList>
```

```
protected void rblStatus_SelectedIndexChanged(object sender,
EventArgs e) {
    RadioButtonList rbl = (RadioButtonList)sender;
    if (rbl.SelectedValue == "S") {
        lblGodinaStudija.Visible = tbGodinaStudija.Visible = true;
    }
    else {
        lblGodinaStudija.Visible = tbGodinaStudija.Visible = false;
    }
}
```

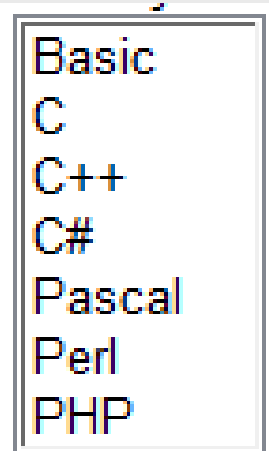
RadioButton naspram *RadioButtonList*

- ❑ Osim kontrole `RadioButtonList` koja sadrži kolekciju elemenata od kojih samo jedan element može biti označen, postoji i kontrola `RadioButton` koja predstavlja izdvojeni element slične funkcionalnosti.
- ❑ Više takvih elemenata se može povezati u jedinstvenu grupu pomoću svojstva `GroupName`
- ❑ Prednost pojedinačnih elemenata su veće mogućnosti dizajna, dok je prednost kontrole `RadioButtonList` jednostavnost korištenja, primjerice kod povezivanja podataka iz nekog izvora (baza podataka, poslovni objekt) kad se upotrebljavaju svojstva:
 - `DataTextField` : stupac iz tablice ili svojstvo poslovnog objekta koje će se koristiti za prikaz teksta pored pojedinog elementa
 - `DataValueFields` : stupac iz tablice ili svojstvo poslovnog objekta koje će se koristiti za postavljanje vrijednost pojedinog elementa
 - `DataSource` ili `DataSourceID` : izvor podataka ili identifikator izvora podataka (ako se radi o nekoj kontroli koja se nalazi na stranici)
 - `DataMember` : ako izvor podataka ima više tablica, označava koja će biti iskorištena za povezivanje

ListBox


- ❑ Sadrži listu elemenata. Svaki ListItem predstavlja jedan element.

```
<asp:ListBox ID="lbxJezici" runat="server" Rows="7"  
SelectionMode="Multiple">  
    <asp:ListItem>Basic</asp:ListItem>  
    <asp:ListItem>C</asp:ListItem>  
    <asp:ListItem>C++</asp:ListItem>  
    <asp:ListItem>C#</asp:ListItem>  
    <asp:ListItem>Pascal</asp:ListItem>  
    <asp:ListItem>Perl</asp:ListItem>  
    <asp:ListItem>PHP</asp:ListItem>  
</asp:ListBox>
```



- ❑ Mogućnost označavanja više elemenata ovisi o vrijednosti svojstva SelectionMode : Single/Multiple
- ❑ Ako lista ima više elemenata nego što je vrijednost svojstva Rows, pojavit će se vertikalni kliznik
- ❑ Događaj koji evidentira promjenu označene opcije: SelectedIndexChanged

Validacijske kontrole


- ❑ Omogućavaju provjeru ispravnosti ulaznih podataka i prije samog *postbacka*
- ❑ Primjer  **Web\Komentar\Default.aspx** koristi dvije validacijske kontrole:
 - RequiredFieldValidator : provjera da li je određeno polje popunjeno
 - RegularExpressionValidator : provjera da li je vrijednost određenog polja u skladu s određenim regularnim izrazom (definiran unutar svojstva `ValidationExpression`)
- ❑ Svaka validacijska kontrola odnosi se na neku kontrolu (svojstvo `ControlToValidate`) te ima tekst poruke (`ErrorMessage`)

```
<asp:RegularExpressionValidator  
ID="RegularExpressionValidator1"      runat="server"  
ControlToValidate="tbGodinaStudija"  
      ErrorMessage="Godina studija mora biti između 1 i 5"  
      ValidationExpression="[1-5]" />
```

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator1"  
runat="server" ControlToValidate="tbKomentar"  
ErrorMessage="Komentar mora biti popunjen" />
```

Primjer validacijskih kontrola

❑ Primjer: Web\Komentar\Default.aspx

Adresa  http://localhost:24887/Komentar/Default.aspx

Popunite obavezne podatke, napišite vlastiti komentar i pošaljite ga autorima ove stranice:

Ja sam:

☒ Student

☐ Promatrač

Trenutna godina studija Godina studija mora biti između 1 i 5

Poznajem sljedeće programske jezike:

Basic

C

C++

C#

Pascal

Perl

PHP

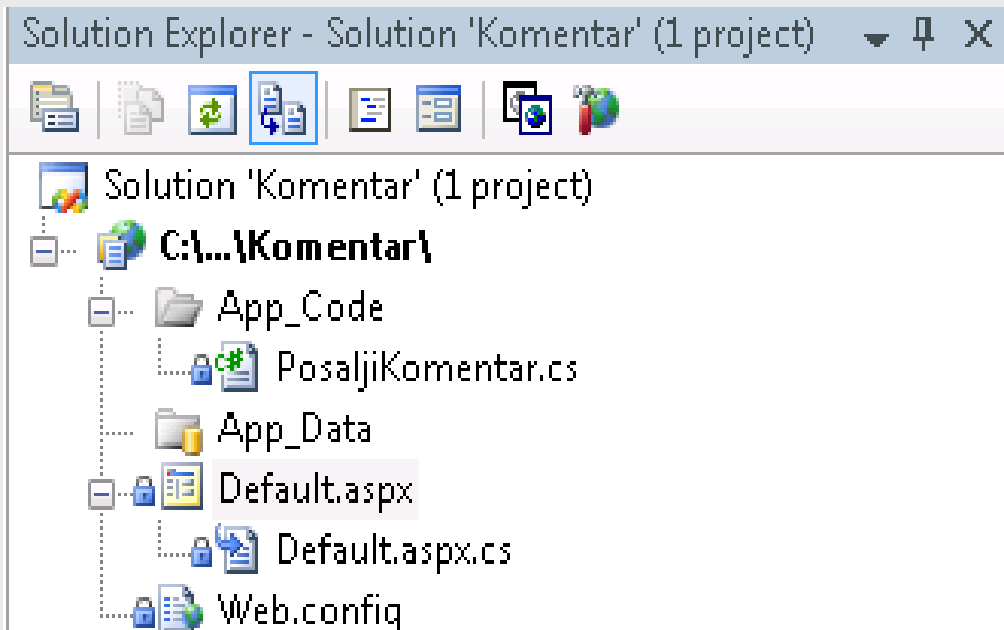
Moj komentar:

Komentar mora biti popunjen

Slanje poruke s komentarom na e-mail

❑ Primjer: Web\Komentar - Default.aspx

- Kod neovisan o pojedinoj stranici smješta se u App_Code kazalo (PosaljiKomentar.cs)
- Postavke za slanje elektroničke pošte (ovisne o smještaju web aplikacije):
 - Adresa mail-servera
 - Adresa pošiljatelja
 - Naslov e-maila
- Postavke se moraju moći lako promijeniti (bez mijenjanja samog kôda)
- Rješenje - web.config datoteka



Konfiguracijska datoteka

❑ Postavke za e-mail nalaze se u datoteci web.config

```
<system.net>
  <mailSettings>
    <smtp from="Boris Milašinović&ltltb>boris.milasinovic@fer.hr</b>";">
      <network host="161.53.18.5"/>
    </smtp>
  </mailSettings>
</system.net>
<appSettings>
  <add key="MailSubject" value="Komentar"/>
</appSettings>
```

- ❑ Korištenjem praznog konstruktora za objekte iz razreda **MailMessage**, odnosno **SmtpClient**, automatski se pune svojstva **From** (adresa pošiljatelja), odnosno svojstva koja definiraju postavke SMTP servera
- ❑ Detaljnije o parametrizaciji i dohvat u postavki iz datoteke **web.config** kasnije (13. poglavlje)

Dohvat postavki i stvaranje e-maila s privitkom

❏ Primjer: Web\Komentar - PosaljiKomentar.cs

```
MailMessage message = new MailMessage();
message.To.Add(message.From.Address);
message.Subject = ConfigurationManager.AppSettings["MailSubject"];
message.IsBodyHtml = false;
message.Body = "Komentar se nalazi u privitku";
message.Body += "\r\nPoslano s adrese: " +
    HttpContext.Current.Request.UserHostAddress;
string tekstKomentara;

...
byte[] buffer = Encoding.Default.GetBytes(tekstKomentara);
using (MemoryStream stream = new MemoryStream(buffer))
{
    Attachment attachment = new Attachment(stream,
        MediaTypeNames.Text.Plain);
    attachment.Name = "komentar.txt";
    message.Attachments.Add(attachment);
    SmtpClient client = new SmtpClient();
    client.Send(message);
}
```

Objavljanje .NET Web aplikacije na Internetu

- ❑ Razvojni poslužitelj (development server) i produkcijski poslužitelj (production server)



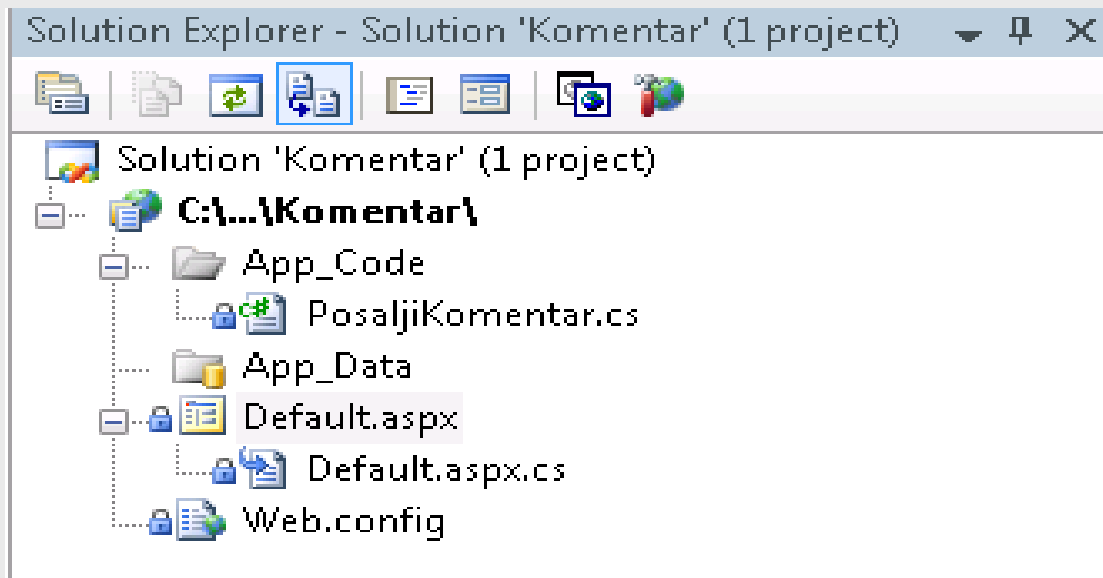
- ❑ Produkcijski poslužitelj

- Stalna veza na Internet
- Windows 200X Server + IIS Server + ASP.NET
- Dozvole potrebne za kreiranje virtualnog foldera, te pristupa resursima
- Mogućnost korištenja iznajmljenog Web prostora - dotNET hosting

- ❑ Datoteke potrebne za izvršavanje aplikacije:

- Priprema datoteka : iz izbornika Build odabrati opciju Publish Web Site
- Datoteke potrebne za izvršavanje: *.aspx, *.ascx, web.config, bin/* (izvorni kod nije potreban i neće biti publiciran)

Dijelovi ASP.NET aplikacije



☐ Publish web site:

/bin

App_Code.dll

App_Web_default.aspx.cdcab7d2.dll

App_Code.compiled

Default.aspx

PrecompiledApp.config

Web.config

Povezivanje podataka u web aplikaciji

Povezivanje podataka

❑ Postupak povezivanja (*Data Binding*)

- kreiranje objekta koji sadrži podatke (DataSet, DataTable, DataView, DataReader, ArrayList, List<>, string[], ..., XML itd.)
 - preciznije, bilo koji razred koji implementira IEnumerable ili *IListSource*
- pridruživanje tog objekta kontroli

❑ Povezivanje navođenjem izraza za povezivanje u ASPX stranici

- `<%# expression %>`
 - `<asp:Label id="Label1" runat="server" Text="<%# 2+3 %>" />`
 - `<asp:label id="Label2" runat="server" Text="<%# TextBox1.Text %>">`

❑ DataBind() – postupak kontrole ili stranice za povezivanje u pozadinskom kodu (.cs datoteci)

- evaluira izraz i obavlja povezivanje
- primjer, povezivanje pojedinačne kontrole (i podkontrola)
 - `Label1.DataBind()`
- primjer, povezivanje svih kontrola na stranici
 - `Page.DataBind()`

Povezivanje podataka (nastavak)

❑ Primjer Web\Komentar - DataBinding.aspx

- Proširenje prethodnog primjera
- Kontrola tipa `Label` se povezuje sa svojstvom

```
<asp:Label id="lblTrenutniDatum" runat="server"
    Text="<%# MyProperty %>"/>

protected string MyProperty {
    get {
        return "Trenutno je: " + DateTime.Now.ToString();
    }
}
```

- Lista jezika (`ListBox`) se povezuje iz kôda s poljem stringova

```
string[] jezici = { "Basic", "C", "C++", "C#", "Pascal",
    "Perl", "PHP" };
lbxJezici.DataSource = jezici;
lbxJezici.DataBind();
```

❑ Promjena početne stranice unutar web aplikacije - desni klik na željenu aspx stranicu i odabir opcije `Set as Start Page`

Povezivanje jednostavnih lista

- ❑ **Pojedini element sadrži vidljivi tekst i opcionalnu vrijednost, koja se ne vidi. (*Primjerice: poštanski broj mjesta kao vrijednost i naziv mjesta kao tekst*)**

- ❑ **Kontrole koje se povezuju na ovaj način**
 - `<asp:ListBox>`
 - `<asp:DropDownList>`
 - `<asp:RadioButtonList>`
 - `<asp:CheckBoxList>`

- ❑ **Postupak povezivanja**
 - opcionalno postavljanje svojstava `DataValueField` i `DataTextField`
 - postavljanje svojstva `DataSource` ili `DataSourceID`
 - opcionalno postavljanje svojstva `DataMember` (ako izvor ima više tablica)
 - poziv postupka `DataBind()`

Primjeri povezivanja lista

❑ Povezivanje samo korištenjem svojstva DataSource


```
string[] jezici= {"Basic", "C", "C++", "C#",  
"Pascal", "Perl", "PHP"};  
lbxJezici.DataSource = jezici;  
lbxJezici.DataBind();
```

❑ Povezivanje korištenjem DataValueField i DataTextField

- povezivanje kontrola obavlja se slično povezivanju na kolekcije
- pretpostavlja se da postoji izvor s poljima, npr. DataView dv
- postavljanje DataSource - DataReader, DataView ili poslovni objekt
- postavljanje DataValueField i DataTextField na polja izvora
- poziv postupka DataBind()

```
lbxJezici.DataSource = dv;  
lbxJezici.DataValueField = "IdJezika";  
lbxJezici.DataTextField = "NazJezika";  
lbxJezici.DataBind();
```

Povezivanje lista s podacima iz baze podataka

- ❑ **Primjer:**  Web\Komentar - DataBindingFromDB.aspx
- ❑ **Postupak** DohvatiJezikeIzBaze vraća **IDataReader** (folija Spajanje ...)
- ❑ **Povezivanje se može definirati:**
 - u kôdu u pozadinskom razredu (.cs)

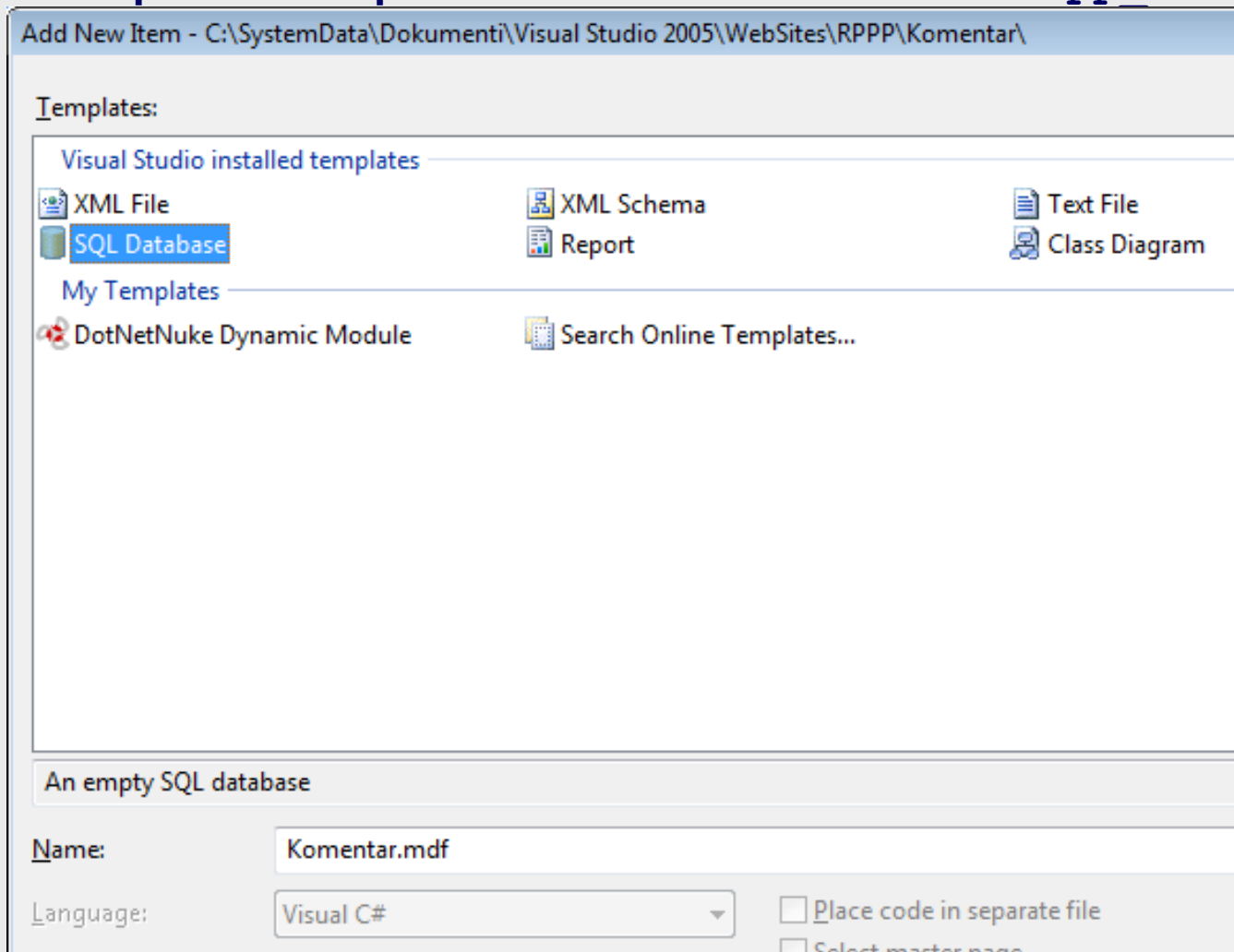
```
void Page_Load(object s, EventArgs e) {  
    using (IDataReader reader = DohvatiJezikeIzBaze()) {  
        lbxJezici.DataSource = reader;  
        lbxJezici.DataTextField = "NazJezika";  
        lbxJezici.DataValueField = "IdJezika";  
        lbxJezici.DataBind();  
    }  
}
```

- odmah pri definiciji kontrole (.aspx) - u kôdu je potrebno pozvati **DataBind**

```
void Page_Load(object s, EventArgs e) {  
    lbxJezici.DataBind();  
}  
  
<asp:ListBox id="lbxJezici" runat="server"  
    DataSource=<%# DohvatiJezikeIzBaze() %>  
    DataValueField="IdJezika"  
    DataTextField="NazJezika" />
```

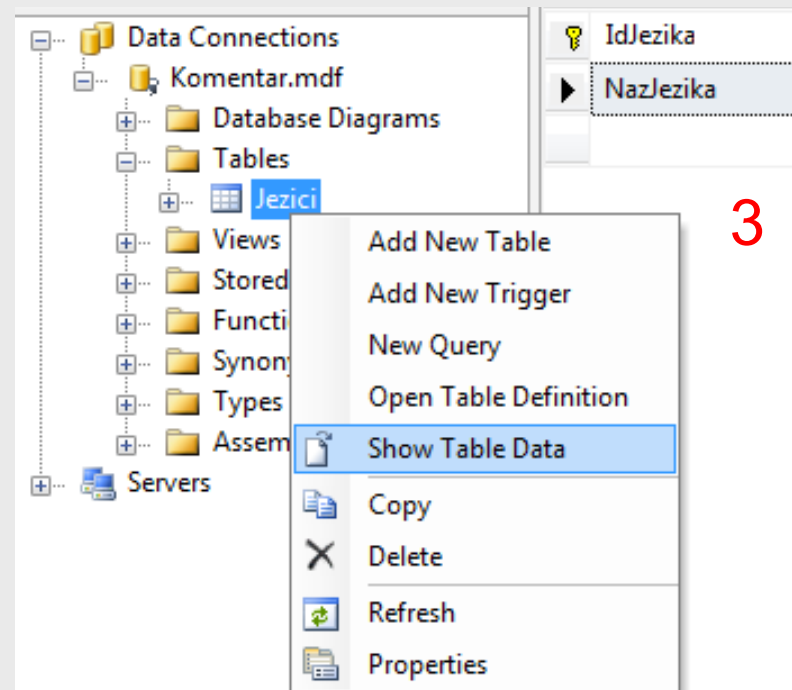
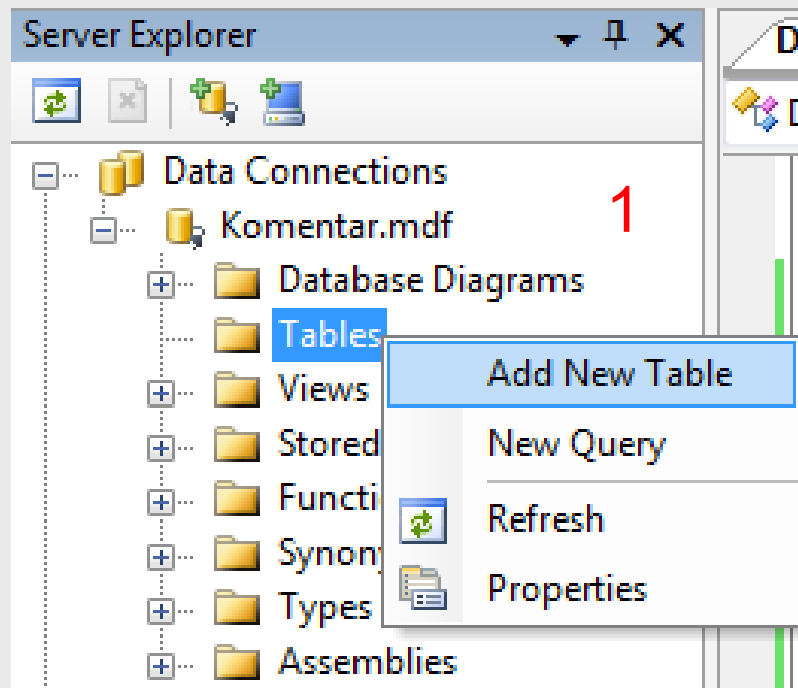
Kreiranje jednostavne baze podataka

- ☐ Dodavanje *SQLExpress* baze podataka unutar postojeće web aplikacije
 - *Add New Item* → *SQL Database*
- ☐ **SQL Express baza podataka nalazit će se u kazalu *App_Data***



Kreiranje objekata u bazi podataka

❑ Primjer: (SQLEXPRESS)



dbo.Jezici: Tabl...TA\KOMENTAR.MDF) DataBinding			
	Column Name	Data Type	Allow Nulls
🔑	IdJezika	int	<input type="checkbox"/>
	NazJezika	varchar(50)	<input type="checkbox"/>

2

Jezici: Query(b...A\KOMENTAR.MDF)		
	IdJezika	NazJezika
	1	Basic
	2	C
	3	C++
	4	C#
	5	Pascal
	6	Perl
	7	PHP

4

Spajanje na bazu podataka

❑ Datoteka web.config : element connectionStrings

```
<connectionStrings>
  <add name="MojaKonekcija"
        connectionString="Data Source = .\SQLEXPRESS;
        AttachDbFilename=|DataDirectory|Komentar.mdf;
        User Instance=True; Integrated Security = true;"/>
</connectionStrings>
```

```
protected IDataReader DohvatiJezikeIzBaze() {
    string connectionString =
        ConfigurationManager.ConnectionStrings[
            "MojaKonekcija"].ConnectionString;
    SqlConnection connection =
        new SqlConnection(connectionString);
    SqlCommand command =
        new SqlCommand("SELECT * FROM Jezik", connection);
    connection.Open();
    return
        command.ExecuteReader(CommandBehavior.CloseConnection);
}
```

Složene liste

GridView, Repeater, ...

Kontrola *GridView*

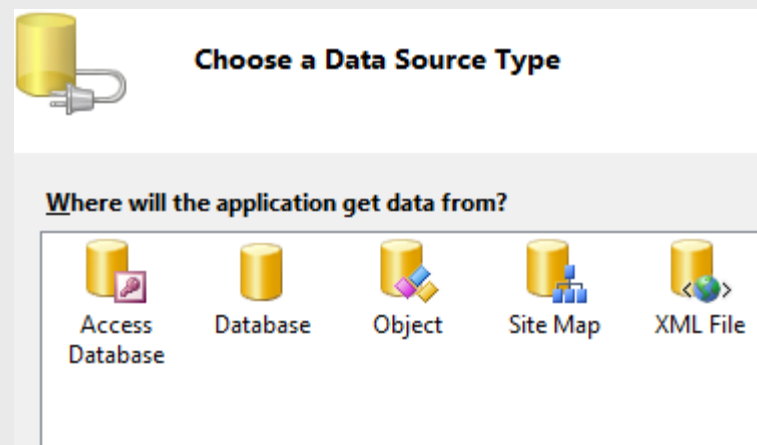
❑ Mreža s podacima

- Standardno prikazuje sve stupce (polja) povezanog izvora, ali se može definirati i podskup stupaca
- Pojedinačni stupac može se formatirati korištenjem predložaka (*templates*)
- Ima svojstva kojima se određuje mogućnost sortiranja podataka, pregleda po stranicama te izmjene i brisanja prikazanih podataka


❑ Povezivanje s podacima

- *GridView* povezuje podatke definirajući izvor podataka (`DataSource`)
 - Access
 - Baza podataka podržana kroz ADO.NET
 - Poslovni objekt
 - Mapa web-sjedišta
 - XML datoteka
- Definiranje izvora podataka:
 - Pri dizajnu
 - U kôdu (tijekom izvođenja) nakon čega slijedi poziv metode `DataBind()`

*`DataBind` nije potreban ukoliko je unutar `aspx` stranice definiran `DataSource`



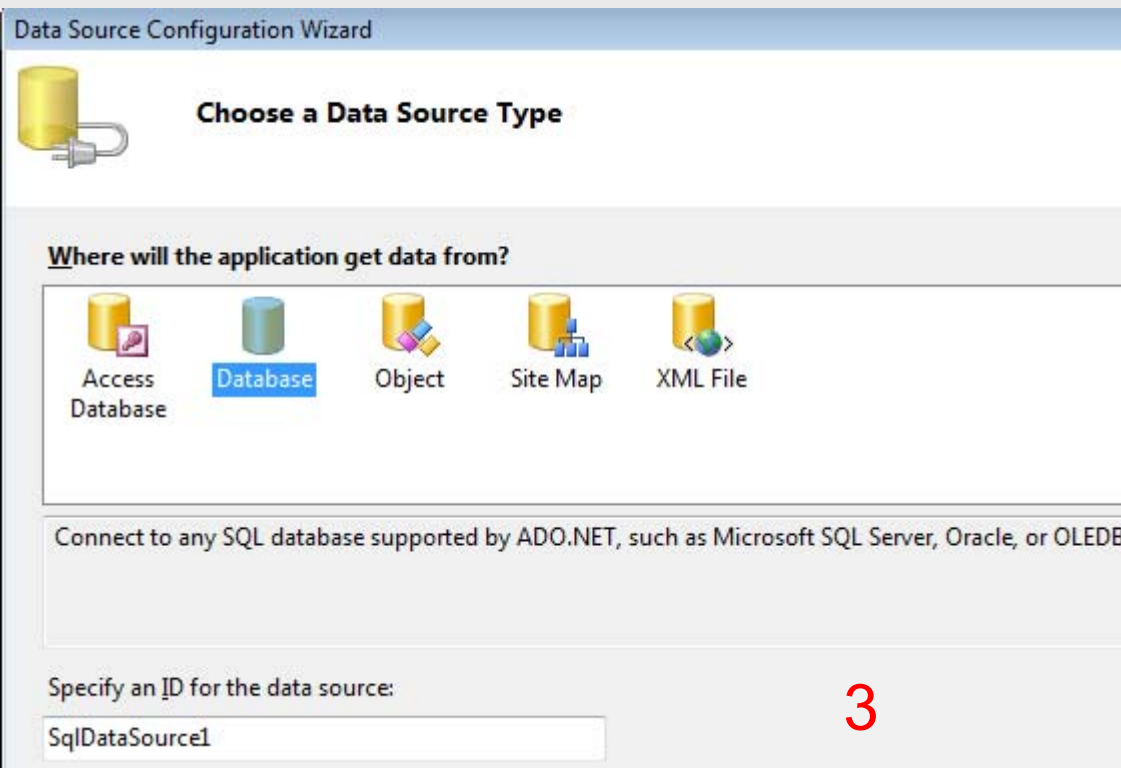
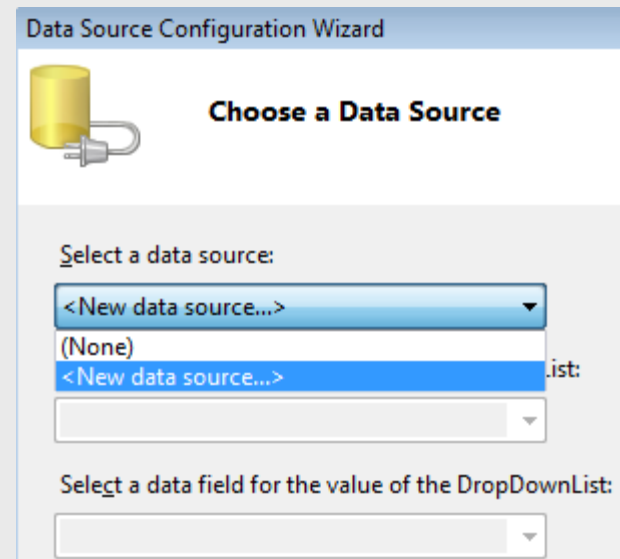
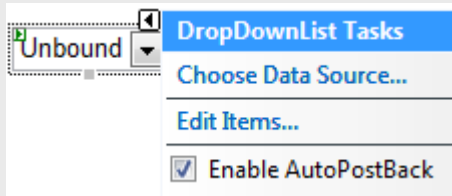
Web stranica oblika zaglavlje-stavke

- ❑ Primjer:  Web\MasterDetail
- ❑ Stranica sadrži padajuću listu s nazivima država i GridView s mjestima u određenoj državi
- ❑ Izvor(i) podataka: SQLDataSource
- ❑ Nakon odabira države unutar padajuće liste, prikazuju se mjesta koja pripadaju navedenoj državi.
- ❑ Jednostavan primjer koji se može realizirati pomoću ugrađenih čarobnjaka za povezivanje



Poštanski broj	Naziv mjesta	Poštanska oznaka mjesta
42225	Butkovec	Breznički Hum
52342	Butkovići	Svetvinčenat
52000	Butoniga	Pazin
10345	Buzadovac	Gradec
51219	Buzdohanj	Čavle
52420	Buzet	Buzet
44401	Buzeta	Hajtić
10010	Buzin	Zagreb-Sloboština
51301	Buzin	Brod na Kupi
53211	Bužim	Smiljan

Postavljanje izvora podataka za padajuću listu



Postavljanje izvora podataka za padajuću listu (2)

Configure Data Source - SqlDataSource1

Choose Your Data Connection

Which data connection should your application use to connect to the database?

New Connection...

Configure Data Source - SqlDataSource1

Save the Connection String to the Application Configuration File

Storing connection strings in the application configuration file simplifies maintenance and deployment. To save the connection string in application configuration file, enter a name in the text box and then click Next. If you choose not to do this, the connection string is saved in the page as a property of the data source control.

Do you want to save the connection in the application configuration file?

☒ Yes, save this connection as:

FirmaConnectionString

< Previous Next > Finish Cancel

Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server (SqlClient) Change...

Server name:
BMPC\SQL2005 Refresh

Log on to the server

☒ Use Windows Authentication
☐ Use SQL Server Authentication

User name: Password: Save my password

Connect to a database


☒ Select or enter a database name:
Firma
☐ Attach a database file:
Browse... Logical name:

Advanced...

Test Connection OK Cancel

Odabir tablice i stupaca za prikaz u padajućoj listi

Configure Data Source - SqlDataSource1

 **Configure the Select Statement**

How would you like to retrieve data from your database?

☐ Specify a custom SQL statement or stored procedure

☒ Specify columns from a table or view

Name:

Columns:

<input type="checkbox"/> *	<input type="checkbox"/> Return only unique rows
<input checked="" type="checkbox"/> OznDrzave	<input type="text" value="WHERE..."/>
<input checked="" type="checkbox"/> NazDrzave	<input type="text" value="ORDER BY..."/>
<input type="checkbox"/> ISO3Drzave	<input type="text" value="Advanced..."/>
<input type="checkbox"/> SifDrzave	

SELECT statement:

< Previous Next > Finish Cancel

Poredak elemenata u listi

❑ Poredak elemenata u listi odabirom opcije ORDER BY

Add ORDER BY Clause

Specify the columns you would like to order by.

Sort by

NazDrzave

☒ Ascending
☐ Descending

Then by

☐ Ascending
☐ Descending

Then by

☐ Ascending
☐ Descending

SELECT statement:


```
SELECT [OznDrzave], [NazDrzave] FROM [Drzava] ORDER BY [NazDrzave]
```

OK Cancel

Određivanje vrijednosti i teksta elemenata

- ❑ Za vrijednost pojedinog elementa u padajućoj listi odabrati stupac OznDrzave, a za tekst odabrati stupac NazDrzave

Data Source Configuration Wizard



Choose a Data Source

Select a data source:

SqlDataSource1 ▼

Select a data field to display in the DropDownList:

NazDrzave ▼

Select a data field for the value of the DropDownList:

OznDrzave ▼

Postavljanje izvora podataka za *GridView*

- ❑ Potrebno je odabrati novi `SQLDataSource`
- ❑ Podaci će biti iz iste baze, ali iz tablice `Mjesto`.
- ❑ Dohvat podataka će biti ograničen samo na ona mjesta koja se nalaze u odabranoj državi
- ❑ Potrebno je ažurirati `WHERE` dio upita

Configure Data Source - SqlDataSource2

Configure the Select Statement

How would you like to retrieve data from your database?

☐ Specify a custom SQL statement or stored procedure

☒ Specify columns from a table or view

Name:
Mjesto

Columns:

- ☐ *
- ☐ IdMjesto
- ☐ OznDrzave
- ☒ NazMjesto
- ☒ PostBrMjesto
- ☒ PostNazMjesto

☐ Return only unique rows

WHERE...

ORDER BY...

Advanced...

SELECT statement:
SELECT [NazMjesto], [PostBrMjesto], [PostNazMjesto] FROM [Mjesto]

< Previous Next > Finish Cancel

Filtriranje podataka temeljem vrijednosti druge kontrole

- ❑ Filtar treba postaviti na stupac OznDrzave i to tako da je njegova vrijednost jednaka vrijednosti odabranog elementa u padajućoj listi

Add WHERE Clause

Add one or more conditions to the WHERE clause for the statement. For each condition you can specify either a literal value or a parameterized value. Parameterized values get their values at runtime based on their properties.

Column: OznDrzave

Operator: =

Source: Control

Parameter properties

Control ID: DropDownList1

Default value:

SQL Expression: [OznDrzave] = @OznDrzave

Value: DropDownList1.SelectedValue

WHERE clause:

SQL Expression	Value

Buttons: Add, Remove, OK, Cancel

Sortiranje i straničenje

- ❑ Kako bi se omogućilo sortiranje i straničenje potrebno je postaviti svojstva `AllowSorting` i `AllowPaging` na `true`
- ❑ Nekoliko kozmetičkih promjena:
 - Ažurirati zaglavlja pojedinih stupaca
 - Odabrati jedan od unaprijed definiranih dizajna za `GridView`
 - Svojstvo `PagerSetting.Mode` postaviti na `NumericFirstLast`



The screenshot shows a web browser window titled "RPPP - Master-Detail - Windows Internet Explorer". The address bar displays "http://localhost:56696/MasterDetail/Default.aspx". Below the browser window, a table is displayed with a dropdown menu set to "Croatia". The table has three columns: "Poštanski broj", "Naziv mjesta", and "Poštanska oznaka mjesta". The table contains 10 rows of data. At the bottom of the table, there is a pagination control showing "61 62 63 64 65 66 67 68 69 70" with navigation arrows.

Poštanski broj	Naziv mjesta	Poštanska oznaka mjesta
42225	Butkovec	Breznički Hum
52342	Butkovići	Svetvinčenat
52000	Butoniga	Pazin
10345	Buzadovac	Gradec
51219	Buzdohanj	Čavle
52420	Buzet	Buzet
44401	Buzeta	Hajtić
10010	Buzin	Zagreb-Sloboština
51301	Buzin	Brod na Kupu
53211	Bužim	Smiljan

<< ... 61 62 63 64 65 66 67 68 69 70 ... >>

Internet trgovina

❑ **Primjer:**  **Web\Artikli**

❑ **Scenarij slučaja korištenja (Kupi proizvod)**

■ **Osnovni scenarij**

- Kupac pregledava Web katalog i stavlja proizvode u košaricu
- Kupac odluči platiti
- Unosi podatke o otpremi
- Sustav prikazuje punu informaciju o cijeni, uključujući otpremu
- Kupac unosi podatke o kreditnoj kartici
- Sustav autorizira kupnju
- Sustav potvrđuje prodaju interaktivno
- Sustav sprema podatke o narudžbi u bazu
- Sustav prikazuje račun i šalje elektroničko pismo potvrde.

■ **Alternativa: Neuspješna autorizacija**

- U koraku 6 sustav ne uspijeva autorizirati karticu => dozvoliti kupcu ponovni unos podataka o kartici

" Kupac pregledava Web katalog ... "

RPPP - Web shop - Windows Internet Explorer

http://localhost:55666/Artikli/Default.aspx

RPPP - Web shop

Artikli u ponudi

Šifra artikla	Naziv	Jedinica mjere	Cijena	
1603	RAM, 512 MB, SDRAM, PC-133, brand name	kom	407,00	Stavi u košaricu
2184	SODIMM PC-133, 256 MB, 144 pins, brand name (for notebooks)	kom	263,00	Stavi u košaricu
2937	Knjiga "PC Škola - Visual Basic 6.0"	kom	100,00	Stavi u košaricu
3194	Knjiga "PC Škola - Office XP"	kom	120,00	Stavi u košaricu
3688	Knjiga "PC Škola - OpenOffice.org 1.0.1"	kom	69,00	Stavi u košaricu
4243	Torba za notebook, MONAS M, crna	kom	283,00	Stavi u košaricu
4409	SODIMM SSSR PC 2700, 256 MB, 200 pins, 333 MHz, brand name (for notebooks) qepod	kom	366,00	Stavi u košaricu
4694	Mrežna kartica wireless CANYON, CN-7106PC, bežična mreža 11 Mbps, PCMCIA	kom	184,00	Stavi u košaricu
4762	USB 2.0 card, CONNEX, 4 port, PCMCIA	kom	199,00	Stavi u košaricu
4763	USB 2.0 card, CONNEX, 2 port, PCMCIA	kom	159,00	Stavi u košaricu

1 2 3 4 5 6 7 8 9 10 ...

Košarica

Šifra artikla	Jedinična mjera	Naziv	Jedinična cijena	Količina	Cijena	
3688		Knjiga "PC Škola - OpenOffice.org 1.0.1"	69,00	1	69,00	Promijeni količinu Izbaci iz košarice
4409		SODIMM SSSR PC 2700, 256 MB, 200 pins, 333 MHz, brand name (for notebooks)	366,00	3	1.098,00	Promijeni količinu Izbaci iz košarice
					1.167,00	

[Plaćanje](#)

Done

Local intranet | Protected Mode: Off

100%

"Kupac odluči platiti ..."

Repeater

RPPP - WebApps - Plaćanje narudžbe - Windows Internet Explorer

http://localhost:55666/Artikli/Placanje.aspx

RPPP - WebApps - Plaćanje narudžbe

[Povratak na odabir artikala](#)

Podaci o dostavi
[Izračun ukupne cijene](#)
[Unos podataka o plaćanju](#)
[Potvrda](#)

Knjiga "PC Škola - OpenOffice.org 1.0.1"	1	69,00	69,00
SODIMM SSSR PC 2700, 256 MB, 200 pins, 333 MHz, brand name (for notebooks)	3	366,00	1.098,00
			Ukupno: 1.167,00

Matični broj:

Ukoliko ste već kupovali možete dohvatiti postojeće podatke za navedeni matični broj

Primatelj:

Adresa računa

Adresa isporuke

E-mail

Način dostave

Ukupan broj narudžbi u RRRP Web Shopu: 68

Ukupan broj narudžbi u RRRP Web Shopu: 68

Local intranet | Protected Mode: Off 100%

koraci
čarobnjaka

" Sustav prikazuje račun ..."

RPPP - WebApps - Plaćanje narudžbe - Windows Internet Explorer

http://localhost:55666/Artikli/Placanje.aspx

RPPP - WebApps - Plaćanje narudžbe

[Povratak na odabir artikala](#)

Vaša narudžba je uspješno obrađena. E-mail s potvrdom narudžbe je poslan na vašu e-mail adresu.

04.02.2008 11:29	Broj računa	2507
Kupac	Boris Milašinović	Matični broj
Adresa računa	Unska 3	
Adresa isporuke	Unska 3 (ZPR)	
Način dostave	Kurirska dostava	

Knjiga "PC Škola - OpenOffice.org 1.0.1"	1	69,00	69,00
SODIMM SSSR PC 2700, 256 MB, 200 pins, 333 MHz, brand name (for notebooks)	3	366,00	1.098,00
Ukupno:			1.167,00
Poštarina:	90,00		
Ukupna cijena:	1257,00		
Kreditna kartica i broj autorizacije:			

Done

Local intranet | Protected Mode: Off

100%

Arhitektura složene web stranice

❑ **Primjer:**  **Web\Artikli**

❑ **Glavni dijelovi aplikacije:**

- `Default.aspx` i `Placanje.aspx` - nasljeđuju izgled `Artikli.master` stranice
- `GridView` - prikaz artikala i sadržaj košarice
 - Artikli se nalaze u bazi podataka
 - Košarica se čuva unutar objekta trenutne sesije
- `Repeater` - prikaz konačnog odabira
- `Wizard` – serverska kontrola za prikupljanje podataka o narudžbi u nekoliko koraka uz jednostavnu navigaciju među koracima
- `Label`, `TextBox`, `DropDownList` kontrole - unos podataka i prikaz iznosa
- `web.config` datoteka s glavnim postavkama aplikacije
 - `ConnectionString`
 - Postavke za slanje e-maila
- Izvor podataka: poslovni objekt (`ObjectDataSource`)

Glavna stranica

❑ Glavna stranica (Master) - ekstenzija *.master*

- Web aplikacija može imati više master stranica
- Predstavlja zajednički izgled (okvir, dizajn) za više stranica
- Uobičajeno sadrži zajedničke elemente (izbornike, zaglavlje, podnožje, ...)

❑ Može sadržavati sve što i obična stranica uz jedan ili više okvira (ContentPlaceholder)

```
<asp:contentplaceholder id="ContentPlaceholder1"
runat="server"/>
```

- Sadržaji okvira će se puniti u pojedinoj stranici
- Stranica koja koristi *master* stranicu, sastoji se samo od okvira koji se puni određenim sadržajem

```
<%@ Page Language="C#" MasterPageFile="~/Artikli.master"
... %>
<asp:Content ID="Content1" Runat="Server"
    ContentPlaceHolderID="ContentPlaceholder1">
    ...
</asp:Content>
```

Glavna stranica

- ❑ Pogled prilikom dizajna na Master.master stranicu (dolje) i Placanje.aspx koja koristi Master.master stranicu (desno)

RPPP - Internet trgovina

ContentPlaceHolder - ContentPlaceHolder1

Ukupan broj narudžbi u RRRP Web Shopu: [lblBrojNarudzbi]

RPPP - Internet trgovina

Content - Content1 (Custom)

[Povratak na odabir artikala](#)

Podaci o dostavi

[Izračun ukupne cijene](#)

Unos podataka o plaćanju

Potvrda

Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound
Ukupno:			Databound

ObjectDataSource - ArtikliObjectDataSource

Matični broj:

Ukoliko ste već kupovali možete dohvatiti postojeće podatke za navedeni matični broj

Primatelj:

Adresa računa

Adresa isporuke

E-mail

Način dostave

Podaci su nepotpuni.

Ukupan broj narudžbi u RRRP Web Shopu: [lblBrojNarudzbi]

Ulančavanje web stranica

- ❑ Wizard – omogućava ulančavanje formi za unos podataka
- ❑ Omogućuje automatsko očuvanje podataka pri navigaciji
- ❑ Tekst kontrola za upravljanje navigacijom može se parametrizirati

```
<asp:Wizard ID="Wizard1" runat="server"
    CancelButtonText="Odustani"
    FinishCompleteButtonText="Potvrdi kupnju"
    FinishPreviousButtonText="Prethodni korak"
    StartNextButtonText="Dalje" StepNextButtonText="Dalje"
    StepPreviousButtonText="Prethodni korak"
    ActiveStepIndex="0"
    OnFinishButtonClick="Wizard1_FinishButtonClick"
    OnNextButtonClick="Wizard1_NextButtonClick">
    <wizardsteps>
        ...
    </wizardsteps>
</asp:Wizard>
```

- ❑ Iza svakog koraka moguće je izvršiti određeni kôd (npr. obrada događaja **OnNextButtonClick** i zaustavljanje prijelaza na sljedeći korak

```
protected void Wizard1_NextButtonClick(object sender,
    WizardNavigationEventArgs e) {
    e.Cancel = true;
```


Čarobnjak

❑ 5 tipova koraka

- **Start** – prvi korak (u navigaciji se pojavljuje opcija za sljedeći korak)
- **Finish** – zadnji korak (pojavljuje se samo opcija za kraj i za prethodni korak)
- **Complete** – korak nakon završetka čarobnjaka (navigacija više nije prisutna)
- **Step** – korak između početka i kraja
- **Auto** – tip koraka se automatski određuje na osnovu poretka

```
<wizardsteps>
  <asp:WizardStep ID="WizardStep2" title="Podaci o dostavi"
    runat="server" StepType="Start"> ... sadržaj koraka ...
</asp:WizardStep>
<asp:WizardStep ID="WizardStep3" title="Izračun ukupne cijene"
  runat="server">
  ...
</asp:WizardStep>
<asp:WizardStep ID="WizardStep4" title="Unos podataka o plaćanju"
  runat="server" StepType="Finish">
  ...
</asp:WizardStep>
<asp:WizardStep ID="WizardStep5" title="Potvrda" runat="server"
  StepType="Complete"> ...
</asp:WizardStep>
</wizardsteps>
```

Mreža s podacima (GridView)

- ❑ Za razliku od primjera *MasterDetail*, gdje je izvor podataka bila baza podataka (SQLDataSource), u ovom primjeru izvor podataka je poslovni objekt (ObjectDataSource)
- ❑ Koriste se prethodno napisani BLL i DAL kompilirani kao dll datoteke koje se referenciraju unutar web aplikacije.

```
<asp:GridView ID="GridViewArtikli" runat="server"
DataKeyNames="SifArtikla"
DataSourceID="ObjectDataSourceArtikli" ...>
```

- ❑ Za izvor podataka potrebno je navesti poslovni objekt koji sadrži postupke za dohvat (SelectMethod), ažuriranje (UpdateMethod), brisanje (DeleteMethod) i dodavanje (InsertMethod) podataka.

```
<asp:ObjectDataSource ID="ObjectDataSourceArtikli" runat="server"
SelectMethod="FetchAll"
TypeName="Firma.ArtiklBllProvider"/>
```

Mreža s podacima – glavna svojstva

❑ Visual Studio – grafičko uređivanje mreže s podacima

- AutoFormat (odabir između nekoliko unaprijed definiranih izgleda mreže)
- EditColumns (grafičko sučelje za definiranje stupaca mreže)

❑ Neka svojstva

- `AutoGenerateColumns` – automatski prikaz svih stupaca iz povezanog izvora
- `AllowSorting`, `AllowPaging` – da li će se prikazati kontrole za sortiranje i straničenje
- Straničenje je najčešće automatski podržano, dok sortiranje, u ovisnosti o izvoru podataka, može zahtijevati pisanje dodatnog kôda
- `DataKeyNames` – kolekcija (primarnih) ključeva, omogućava jednoznačno određivanje pojedinog retka mreže prilikom brisanja i ažuriranja
 - npr: primarni ključ u prikazu artikala je `IdArtikla`
 - pojedinom ključu unutar kôda se pristupa pomoću kolekcije `DataKeys`
npr: `grid.DataKeys[index].Value`
- `DataSource` – izvor podataka kojim će se mreža popuniti

Definiranje stupaca u mreži s podacima

- ❑ **Potrebno je postaviti vrijednost svojstva `AutoGenerateColumns` na `false`**
- ❑ **Svojstvo `Columns` sadrži definicije stupaca koji mogu biti različitih tipova:**
 - `BoundField` - vezani stupac. Stupac se povezuje sa stupcem u izvoru podataka navođenjem imena stupca u svojstvu `DataField`.
 - `TemplateField` - stupac definiran preko predložaka
 - `HyperLinkField` - stupac u kojem se nalaze poveznice
 - `CheckBoxField` - stupac u kojem su elementi tipa `CheckBox`
 - `ImageField` - stupac koji sadrži slike
 - `ButtonField` - stupac koji sadrži gumbе i predstavljaju proizvoljnu naredbu za mrežu s podacima. Sam gumb može biti prikazan kao poveznica koja izaziva `PostBack` (`LinkButton`), gumb u obliku slike (`ImageButton`) ili kao klasični gumb (`Button`).
 - `CommandField` - stupac čiji elementi mogu biti gumb za označavanje (`Select`), gumb za brisanje (`Delete`) i tri gumba za ažuriranje pojedinog retka (`Edit`, `Update`, `Cancel`). Tekst pojedinog gumba može se promijeniti

Povezivanje pojedinačnih stupaca

- ❑ Postavlja se `AutoGenerateColumns="false"`
- ❑ Svojstvo `Columns` sadrži definicije stupaca
- ❑ Stupci su vezani (`BoundField`) uz određeno polje iz baze (`DataField`)

```
<asp:GridView ID="GridViewArtikli" runat="server"
AutoGenerateColumns="False" DataKeyNames="SifArtikla"
DataSourceID="ObjectDataSourceArtikli" AllowPaging="True"
OnRowCommand="GridView1_RowCommand"
OnRowDataBound="GridViewArtikli_RowDataBound">
    <Columns>
        <asp:BoundField DataField="SifArtikla" HeaderText="Šifra
        artikla" /> ...
        <asp:BoundField DataField="CijArtikla"
            DataFormatString="{0:00}" HeaderText="Cijena"/>
        <asp:ButtonField ButtonType="Button" CommandName="Stavi"
            Text="Stavi u košaricu" />
    </Columns>
</asp:GridView>
<asp:ObjectDataSource ID="ObjectDataSourceArtikli" runat="server"
    SelectMethod="FetchAll" TypeName="Firma.ArtiklBllProvider">
</asp:ObjectDataSource>
```

Povezivanje pojedinačnih stupaca (2)

- ❑ **Ograničena funkcionalnost vezanih stupaca → predlošci za stupce**
 - Pri dizajnu stranice klik na gornji desni rub mreže s podacima -> `Edit Columns` → klik na stupac → "Convert this field into a `TemplateField`"
- ❑ **Moguće definirati izgled:**
 - zaglavlja stupca (`HeaderTemplate`)
 - podnožja stupca (`FooterTemplate`)
 - pojedinog retka (`ItemTemplate`)
 - svakog drugog retka (`AlternatingItemTemplate`)
 - retka prilikom ažuriranja (`EditItemTemplate`)
 - novog retka u mreži (`InsertItemTemplate`)
- ❑ **Kod stupaca definiranih preko predložaka potrebno je eksplicitno povezati element stupca mreže sa stupcem u izvoru (*Eval, Bind*)**

Povezivanje pojedinačnih stupaca (3)

❑ **DataRowBinder** – statički objekt za povezivanje

- Eval – postupak za dohvat i formatiranje podataka u predlošku
 - `<%# Eval("Cijena", "{0:N2}") %>`

❑ **Podacima se može pristupiti i drugačije u ovisnosti o tipu podataka koji se prikazuju pomoću mreže**

- DataTable
 - `<%# ((DataRow)Container.DataItem) ["FieldName"] %>`
- DataView
 - `<%# ((DataRowView)Container.DataItem) ["FieldName"] %>`
- List<string>
 - `<%# ((string)Container.DataItem) %>`

❑ **Dvosmjerno povezivanje (koristi se kad je potrebno izvesti ažuriranje pojedinih redaka u mreži s podacima)**

- Bind
 - `<%# Bind("Cijena", "{0:N2}") %>`

Povezivanje pojedinačnih stupaca (4)

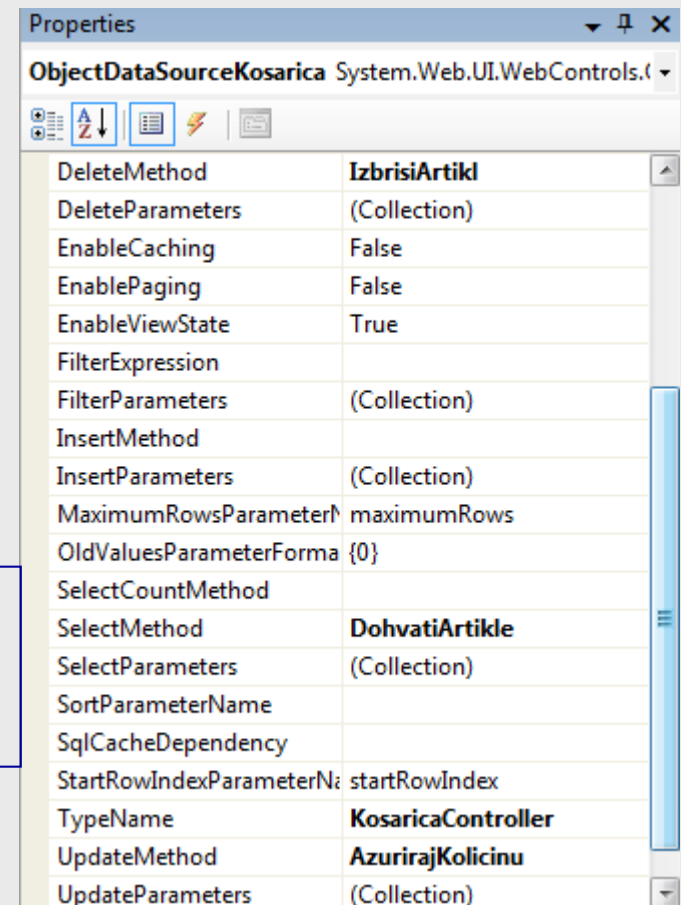
- ❑ Limitiranost vezanih stupaca => stupci u obliku predloška (TemplateField)
- ❑ Stupci s gumbima za ažuriranje i brisanje retka iz mreže s podacima (CommandField)

```
<Columns>
  <asp:TemplateField HeaderText="Cijena" ItemStyle-HorizontalAlign="Right">
    <ItemTemplate>
      <asp:Label ID="Label1" runat="server" Text='<%# Eval("Iznos",
        "{0:N2}") %>'></asp:Label>
    </ItemTemplate>
    <FooterTemplate>
      <asp:Label ID="lblTotal" runat="server"
        Text='<%# KosaricaController.Ukupno().ToString("N2") %>' />
    </FooterTemplate>
  </asp:TemplateField>
  <asp:CommandField ButtonType="Button"
    CancelText="Odustani"
    EditText="Promijeni količinu"
    ShowEditButton="True"
    UpdateText="Spremi" />
  <asp:CommandField ShowDeleteButton="True" DeleteText="Izbaci iz košarice"
    ButtonType="Button" />
</Columns>
```


Ažuriranje redaka u mreži s podacima

- ❑ "Gotova" funkcionalnost – za ažuriranje se brine izvor podataka
- ❑ Izvor podataka definira postupke za dohvat (`SelectMethod`), dodavanje (`InsertMethod`), ažuriranje (`UpdateMethod`), brisanje (`DeleteMethod`)
- ❑ Parametri postupaka moraju imati iste nazive kao nazivi stupaca koji su vezani u mreži
- ❑ *Čarobnjak za povezivanje izvora podataka i metoda za rad s podacima bit će jednostavniji ukoliko se iznad postupka stavi posebni atribut koji sugerira što postupak radi (`Delete`, `Fill`, `Insert`, `Select`, `Update`)

```
[DataObjectMethod(DataObjectMethodType.Select)]
public static List<Stavka> DohvatiArtikle()
{
    ...
}
```



The screenshot shows the Properties window for an `ObjectDataSourceKosarica` control. The table below represents the data visible in the window:

Property	Value
<code>DeleteMethod</code>	<code>IzbrisiArtikl</code>
<code>DeleteParameters</code>	(Collection)
<code>EnableCaching</code>	False
<code>EnablePaging</code>	False
<code>EnableViewState</code>	True
<code>FilterExpression</code>	
<code>FilterParameters</code>	(Collection)
<code>InsertMethod</code>	
<code>InsertParameters</code>	(Collection)
<code>MaximumRowsParameterName</code>	<code>maximumRows</code>
<code>OldValuesParameterFormName</code>	<code>{0}</code>
<code>SelectCountMethod</code>	
<code>SelectMethod</code>	<code>DohvatiArtikle</code>
<code>SelectParameters</code>	(Collection)
<code>SortParameterName</code>	
<code>SqlCacheDependency</code>	
<code>StartRowIndexParameterName</code>	<code>startRowIndex</code>
<code>TypeName</code>	<code>KosaricaController</code>
<code>UpdateMethod</code>	<code>AzurirajKolicinu</code>
<code>UpdateParameters</code>	(Collection)

Ažuriranje redaka u mreži s podacima

- ☐ Pri pridruživanju postupka koji će se izvršiti za određenu akciju na mreži s podacima (brisanje, ažuriranje) potrebno je navesti parametre.
- ☐ Vrijednost pojedinog parametra jednaka je vrijednosti konkretnog elemenata u stupcu s istim imenom kao što je ime parametra.

Ažuriranje redaka u mreži s podacima (2)

```
<asp:ObjectDataSource ID="ObjectDataSourceKosarica" runat="server"
SelectMethod="DohvatiArtikle" DeleteMethod="IzbrisiArtikl"
    TypeName="KosaricaController" UpdateMethod="AzurirajKolicinu" >
    <UpdateParameters>
        <asp:Parameter Name="KolArtikla" Type="Int32" />
        <asp:Parameter Name="SifArtikla" Type="Int32" />
    </UpdateParameters>
    <DeleteParameters>
        <asp:Parameter Name="SifArtikla" Type="Int32" />
    </DeleteParameters>
</asp:ObjectDataSource>
```

```
public static void AzurirajKolicinu(int KolArtikla, int SifArtikla)
{
    List<Stavka> kosarica
        = HttpContext.Current.Session["Kosarica"] as List<Stavka>;
    if (kosarica != null) {
        for (int i = 0; i < kosarica.Count; i++) {
            if (kosarica[i].SifArtikla == SifArtikla) {
                kosarica[i].KolArtikla = KolArtikla; return;
            }
        }
    }
}
```

Događaji unutar mreže s podacima

□ Događaji na mreži s podacima za koje se može napisati vlastiti rukovatelj događaja:

- DataBound – nastaje nakon što su podaci povezani s mrežom
- PageIndexChanged – nastaje nakon što je mreža s podacima obradila zahtjev za promjenu stranice
- PageIndexChanged – nastaje nakon klika na gumb za promjenu trenutne stranice unutar mreže s podacima, a prije nego mreža s podacima sama obradi događaj
- RowCancelingEdit – nastaje pri ažuriranju retka nakon klika na gumb `Cancel`, a prije nego se izađe iz načina rada koji omogućava ažuriranje
- RowCommand – nastaje nakon klika na gumb koji se nalazi unutar mreže s podacima
- RowCreated – nastaje nakon što se stvori pojedini redak u mreži
- RowDataBound – nastaje nakon što se pojedini redak poveže s podacima iz izvora
- RowDeleted – nastaje nakon brisanja pojedinog retka
- RowDeleting – nastaje pri brisanju retka, nakon klika na gumb `Delete`, a prije samog brisanja
- RowEditing – nastaje pri zahtjevu za ažuriranje retka mreže nakon klika na gumb `Edit`, a prije nego samog ulaza u način rada za ažuriranje
- RowUpdated – nastaje nakon ažuriranje pojedinog retka
- RowUpdating – nastaje nakon klika na gumb `Update`, ali prije samog ažuriranja
- SelectedIndexChanged – nastaje pri promjeni označenog retka
- SelectedIndexChanging – nastaje nakon klika na gumb `Select`, prije promjene označenog retka
- Sorted – nastaje nakon što je mreža s podacima sortirana po određenom stupcu
- Sorting – nastaje klikom na stupca po kojem se želi sortirati, a prije samog sortiranja
- DataBinding, Disposed, Init, Load, Unload, PreRender – naslijeđeno iz razreda `Control`

Vlastite naredbe unutar mreže s podacima

- ☐ Na mreži s podacima se mogu definirati i proizvoljne naredbe, primjerice naredba "Stavi" vezana uz gumb kojim se pojedini artikl stavlja u košaricu
- ☐ Kako bi mreža s podacima reagirala na proizvoljne naredbe, potrebno je definirati postupak koji će obraditi pojavu događaja `RowCommand` koji nastaje klikom bilo kojeg gumba unutar mreže s podacima
- ☐ Naziv naredbe koja je aktivirala navedeni događaj se nalazi u svojstvu `CommandName`

Vlastite naredbe unutar mreže s podacima (2)

```
<asp:GridView ... OnRowCommand="GridView1_RowCommand" ...>
<Columns>...
    <asp:ButtonField ButtonType="Button"
        CommandName="Stavi" Text="Stavi u košaricu" />
</Columns>
```

```
protected void GridView1_RowCommand(object sender,
GridViewCommandEventArgs e){
    if (e.CommandName == "Stavi")
    {
        GridView gridView = e.CommandSource as GridView;
        int Rbr = int.Parse(e.CommandArgument.ToString());

        //dohvati vrijednost primarnog ključa za ovaj redak
        int SifArtikla = (int)gridView.DataKeys[Rbr].Value;
        KosaricaController.DodajUKosaricu(SifArtikla);

        //osvjezi košaricu
        GridViewKosarica.DataBind();
        hlPlacanje.Visible = GridViewKosarica.Rows.Count > 0;
    }
}
```

Proširenje standardne funkcionalnosti

- ❑ Cijenu pojedinog artikla obojiti crveno, plavo ili zeleno u ovisnosti u trenutnoj vrijednosti retka
 - događajem obrade dodavanja podataka u redak mreže (RowDataBound)

```
<asp:GridView ID="GridViewArtikli" runat="server" ...  
    OnRowDataBound="GridViewArtikli_RowDataBound" ...>
```

```
protected void GridViewArtikli_RowDataBound(  
    object sender, GridViewRowEventArgs e) {  
    if (e.Row.RowType == DataControlRowType.DataRow)  
    {  
        //dohvati cijenu artikla  
        Artikl artikl = (Artikl) e.Row.DataItem;  
        decimal cijena = artikl.CijArtikla;  
        //cijena je četvrta po redu  
        TableCell cell = e.Row.Cells[3];  
        if (cijena < 150)  
            cell.ForeColor = System.Drawing.Color.Green;  
        else if (cijena < 300)  
            cell.ForeColor = System.Drawing.Color.Blue;  
        else  
            cell.ForeColor = System.Drawing.Color.Red;  
    }  
}
```

Predlošci (templates)

❑ Prilagodba izgleda serverske kontrole

- oblikuje se struktura, a ne samo stil
- u predlošku se mogu koristiti druge kontrole
 - predlošci određuju skup *markup* (HTML ili server) kontrola za svaki stupac

❑ Standardni predlošci za *list-bound* kontrole

- HeaderTemplate: prikazuje se jednom, prije prikaza redaka
- ItemTemplate: prikazuje se za svaki prikazani redak
- AlternatingItemTemplate: slično ItemTemplate, ali za svaki drugi redak
- SeparatorTemplate: prikazuje se između susjednih redaka
- FooterTemplate: prikazuje se jednom, nakon prikaza redaka



Jednostavna lista elemenata (Repeater)

❑ Primjer: ispis računa iz prethodnog primjera

- Dizajn se obavlja ručno, pisanjem HTML predložaka (templates)
- Ne podržava pregled po stranicama ni editiranje podataka
- Prikladan za prikaz podataka bez ažuriranja ("lakši" u odnosu na GridView)

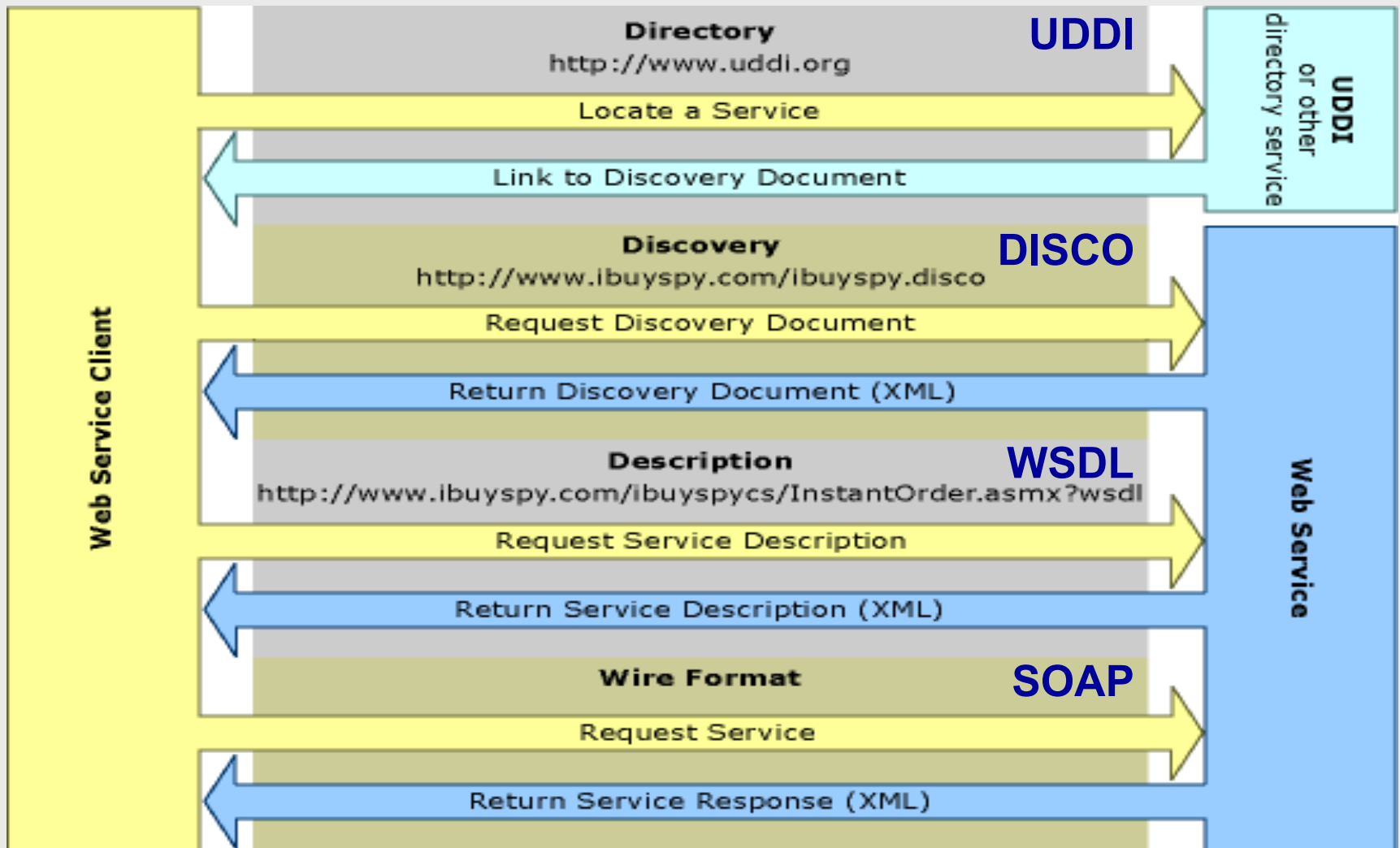
```
<asp:Repeater ID="KosaricaRepeater" runat="server"
    DataSourceID="ArtikliObjectDataSource">
<HeaderTemplate><table border="1" width="100%"></HeaderTemplate>
<ItemTemplate>
    <tr><td><%# Eval("NazArtikla") %></td><td><%#
Eval("KolArtikla") %></td>
    <td align="right"><%# Eval("JedCijArtikla") %></td>
    <td align="right"><%# Eval("Iznos", "{0:N2}") %></td></tr>
</ItemTemplate>
<FooterTemplate>
    <tr><td colspan="3" align="right">Ukupno:</td>
    <td><%# KosaricaController.Ukupno().ToString("N2")
%></td></tr></table>
</FooterTemplate>
</asp:Repeater>
```

Web servisi

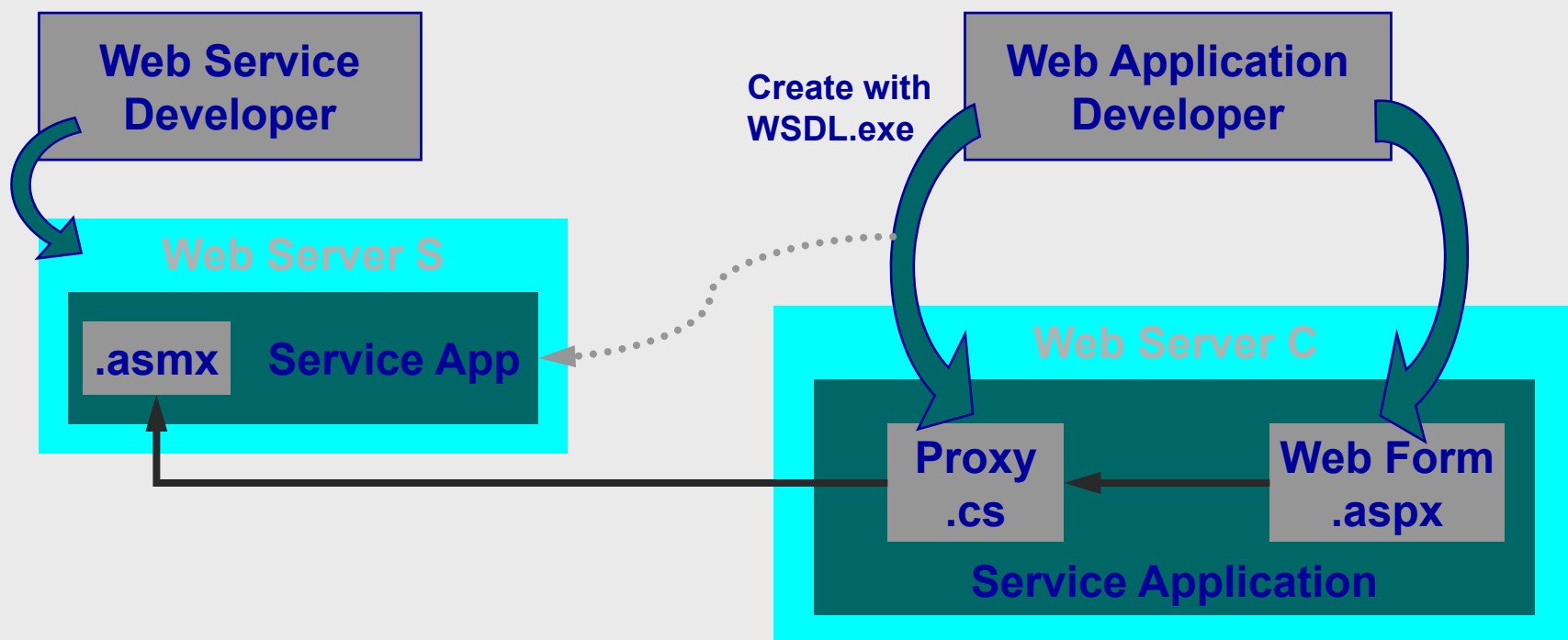
Web servis (Web Service)

- ❑ **Web servis – programabilni URI (Universal Resource Identifier)**
 - Osigurava mehanizam za pozivanje udaljenih postupaka
 - Crne kutije
 - Koristi Web standarde HTTP, XML
- ❑ **SOAP – Simple Object Access Protocol**
 - laki protokol za razmjenu informacija u distribuiranim, heterogenim okruženjima
 - SOAP \approx HTTP + XML (zapravo XML preko HTTP-a)
- ❑ **WSDL - Web Services Description Language**
 - XML shema za opis Web servisa
 - definira format postupaka koje pruža Web servis
- ❑ **UDDI - Universal Description, Discovery, and Integration**
 - Discovery – postupak lociranja Web servisa koji se nalaze na nekom URL-u
 - Web site objavljuje DISCO dokumente, koji vraćaju URL i WSDL opise
 - <http://uddi.xml.org> - Registracijska baza Web servisa

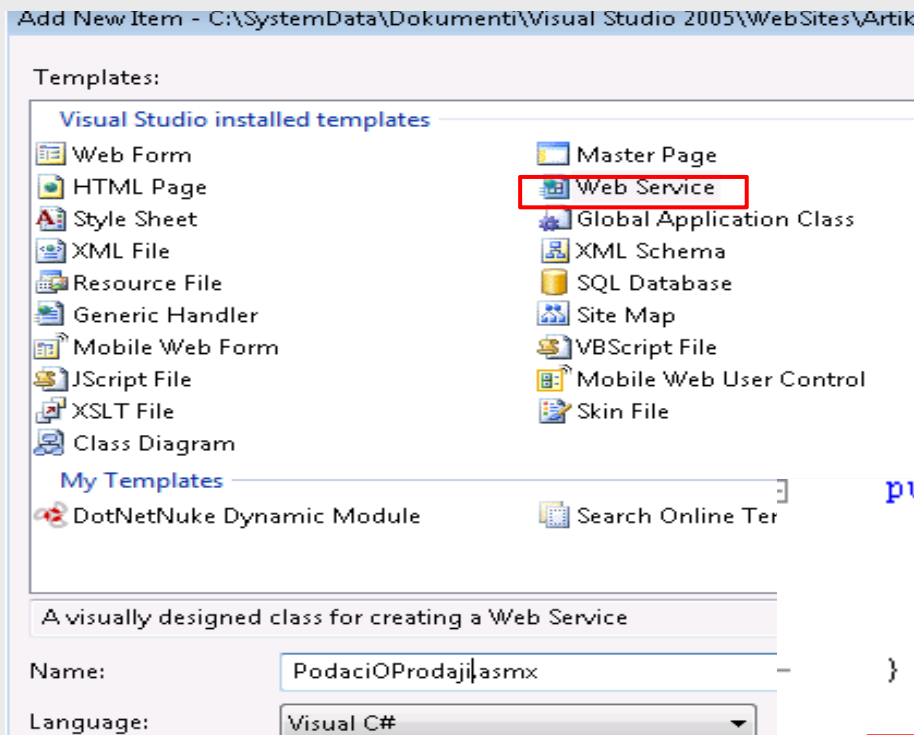
Web servisi



Korištenje Web servisa



Primjer Web Servisa : podaci o prodaji



☐ Odvajanje dizajna i koda

- PodaciOProdaji.asmx
- PodaciOProdaji.cs

☐ Postupci koji se objavljuju ispred imena metode imaju atribut **[WebMethod]**

```
public PodaciOProdaji () {  
  
    // Uncomment the following line if us:  
    // InitializeComponent();  
  
}  
  
[WebMethod]  
public int BrojNarudzbi() {  
    string connectionString = Configuration  
    using (SqlConnection connection = new S  
        SqlCommand command = new SqlCommand(''  
        command.CommandType = CommandType.Te  
        connection.Open();  
        int brojNarudzbi = int.Parse(command  
        return brojNarudzbi;  
    }  
}
```

Referenciranje web servisa unutar projekta

❑ Desni klik na projekt -> Add Web Reference

Add Web Reference

Navigate to a web service URL and click Add Reference to add all the available service

Back Forward Stop Home

URL:

Start Browsing for Web Services

Use this page as a starting point to find Web services. You can click the links below, or type a known URL into the address bar.

Browse to:

- Web services in this solution
- Web services on the local machine
- Browse UDDI Servers on the local network
Query your local network for UDDI servers.

Web Services in this Solution

The Web services available in this solution are listed below. Click the service link to browse that service.

Services	Project	URL
AutorizacijaKartica	C:\...\ArtikliWebService\	AutorizacijaKartica.asmx
PodaciOProdaji	C:\...\ArtikliWebService\	PodaciOProdaji.asmx

URL:

PodaciOProdaji

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [BrojNarudzbi](#)

Web services found at this U

1 Service Found:

- PodaciOProdaji

Web reference name:

Solution 'Artikli' (2 projects)

- C:\...\Artikli\
- App_Code
 - Artikl.cs
 - ArtiklController.cs
 - DokumentController.cs
 - KosaricaController.cs
 - Partner.cs
 - PartnerController.cs
- App_Data
- App_WebReferences
 - AuthorizationWebService
 - AutorizacijaKartica.disco
 - AutorizacijaKartica.discomap
 - AutorizacijaKartica.wsdl
 - PodaciWebService
 - PodaciOProdaji.disco
 - PodaciOProdaji.discomap
 - PodaciOProdaji.wsdl
- Artikli.master
 - Artikli.master.cs

Primjer Web Servisa

❑ Primjer: Web\ArtikliWebService

- <http://localhost:56453/ArikliWebService/PodaciOProdaji.asmx>
- <http://localhost:56453/ArikliWebService/PodaciOProdaji.asmx?WSDL>

❑ Dodavanjem reference na web servis u projekt se dodaju

- ...\\PodaciOProdaji.wsd
- ...\\PodaciOProdaji.disco
- ...\\PodaciOProdaji.discomap

❑ Pozivanje referenciranog web servisa iz kôda

```
PodaciWebService.PodaciOProdaji podaci =  
    new PodaciWebService.PodaciOProdaji();  
lblBrojNarudzbi.Text = podaci.BrojNarudzbi().ToString();
```


Primjer web servisa

❑ Web servis može imati i složeniji ulaz i izlaz

- Primjer:  Web\ArtikliWebService\AutorizacijaKartica.asmx
- Ulaz: razred *PodaciOKartici*
- Izlaz: razred *RezultatAutorizacije*

```
[WebMethod]  
public RezultatAutorizacije Autorizacija(PodaciOKartici podaci)
```

❑ Razredi za komunikaciju definirani su unutar web servisa (a ne klijenta).

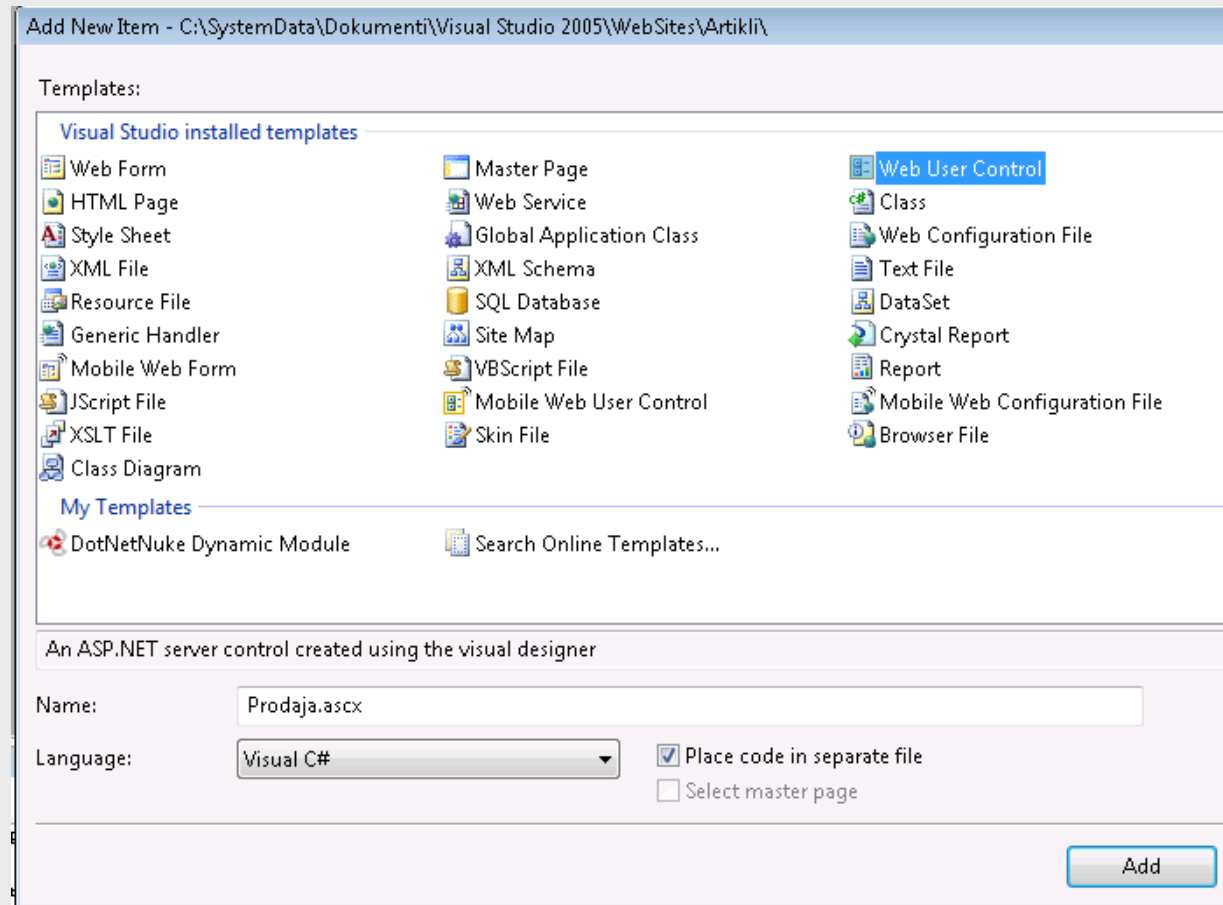
❑ Dodavanjem web reference Visual Studio će automatski generirati omotače onih razreda koje se koriste u komunikaciji.

- Generirani razredi na klijentu sadržavat će sva ona svojstva koja su unutar odgovarajućeg razreda na web servisu javna i namijenjena i za čitanje i za pisanje.
- Postupci definirani u razredima na web servisu ne pojavljuju se u generiranim razredima.

Korisničke kontrole

Korisničke kontrole

- ❑ Složene kontrole koje mogu sadržavati ostale kontrole, a imaju efekt mini-stranica
- ❑ Ekstenzija `ascx`
- ❑ Razdvajanje koda od dizajna:
(`ascx` i `ascx.cs`)



Korisničke kontrole

- ❑ **Primjer:**  **Web\Artikli - Prodaja.ascx**
 - Kontrola za prikaz ukupnog broja narudžbi

- ❑ **Mogu imati javna svojstva**

```
private Color labelColor;
public Color LabelColor {
    get { return labelColor; }
    set { labelColor = value; }
}
```

- ❑ **Na stranici na kojoj ih se koristi obavlja se registracija kontrole**

```
<%@ Register Src="Prodaja.ascx" TagName="Prodaja" TagPrefix="uc1" %>
```

- ❑ **Pri upotrebi koristi se prethodno registrirani prefix (npr. uc1)**

```
<uc1:Prodaja id="Prodaja1" runat="server" LabelColor="Green"/>
```

- ❑ **Po potrebi se može dinamički dodati unutar kôda**

```
Control prodaja = Page.LoadControl("Prodaja.ascx");
((Prodaja) prodaja).LabelColor = Color.Red;
Panel1.Controls.Add(prodaja);
```

Provjera autentičnosti i autorizacija korisnika

Provjera autentičnosti korisnika

- ❑ **Način provjere autentičnosti (autentifikacija) korisnika postavlja se u datoteci `web.config`**

```
<authentication mode="način provjere"/>
```

- ❑ **Četiri načina provjere autentičnosti:**


- **Windows:** Provjera se prenosi na IIS te se koriste korisnički računi na lokalnom računalu i oni iz domene kojoj računal pripada. Pri otvaranju zaštićene stranice pojavljuje se standardni prozor za unos korisničkog imena i zaporke (najčešće NTLM, kao kod dijeljenja datoteka, spajanja na TFS i slično)
- **Forms:** Provjera se vrši unošenjem korisničkog imena i zaporke na određenoj aspx stranici. Podaci o korisnicima se u tom slučaju nalaze u nekoj proizvoljnoj bazi.
- **Passport:** Koristi Microsoft Password Network uslugu za ovjeru korisnika
- **None:** Nema provjere korisnika, aplikacija radi samo s anonimnim korisnicima

Autorizacija korisnika

- ❑ **Pristup pojedinoj stranici ili svim stranicama ispod određenog kazala može se postaviti pomoću `location` oznake unutar datoteke `web.config`**
- ❑ **Primjer: Dozvola pristupa svim stranicama čiji je put oblika `http://server/Admin/stranica.aspx` svim korisnicima iz administratorske grupe na lokalnom računalu te lokalnom korisniku `boris` i korisniku `ivana` iz FER domene.**

```
<configuration>
<location path="Admin">
  <system.web>
    <authorization>
      <allow roles="Administrators" />
      <allow users="boris,FER\ivana" />
      <deny users="*" />
    </authorization>
  </system.web>
</location>
</configuration>
```

Primjer provjere autentičnosti kroz formu

- ❑ **Primjer:**  **Web\Artikli**
- ❑ **Stranica Admin\Default.aspx bit će dostupna samo grupi korisnika pod nazivom Dobavljac**
- ❑ **Provjera autentičnosti vrši se pomoću formi i tu svrhu je dodana stranica Login.aspx i na nju dodana kontrola <asp:Login>**

```
<authentication mode="Forms">  
    <forms loginUrl="Login.aspx"/>  
</authentication>
```

```
<configuration>  
<location path="Admin/Default.aspx">  
    <system.web>  
        <authorization>  
            <allow roles="Dobavljac" />  
            <deny users="*" />  
        </authorization>  
    </system.web>  
</location>  
</configuration>
```



Primjer provjere autentičnosti kroz formu

- ☐ Ugrađena ASP.NET kontrola `Login` provjeru autentičnosti vrši pomoću jednog od definiranih pružatelja usluga provjere (eng. `Provider`)
- ☐ `Login` kontrola sadrži svojstva koja omogućuju promjenu njenog izgleda kao i promjenu teksta koji se prikazuje (`LoginButtonText`, `FailureText`, `TitleText`, `UserNameLabelText`, ...)
- ☐ Ukoliko drugačije nije definirano, koristi se `AspNetSqlProvider` za provjeru korisnika i `AspNetSqlRoleProvider` za rad s grupama korisnika dok se kao baza korisnika koristi SQL Express baza podataka `ASPNET.MDF` u kazalu `App_Data`
- ☐ Prilikom razvoja aplikacije može se koristiti ugrađeni administracijski alat koji se može pokrenuti iz izbornika `Website` odabirom opcije `ASP.NET Configuration`
- ☐ Prvo pokretanje administracijskog alata stvorit će praznu bazu podataka i smjestiti je u kazalo `App_Data`

Ugrađeni administracijski alat za web aplikacije

ASPMicrosoft**.net**

Web Site Administration Tool

How do I use this tool? 

HomeSecurityApplicationProvider

You can use the Web Site Administration Tool to manage all the security settings for your application. You can set up users and passwords (authentication), create roles (groups of users), and create permissions (rules for controlling access to parts of your application).

By default, user information is stored in a Microsoft SQL Server Express database in the Data folder of your Web site. If you want to store user information in a different database, use the Provider tab to select a different provider.

[Use the security Setup Wizard to configure security step by step.](#)

Click the links in the table to manage the settings for your application.

Users	Roles	Access Rules
Existing users: 0 Create user Manage users Select authentication type	Roles are not enabled Enable roles Create or Manage roles	Create access rules Manage access rules

Ugrađeni administracijski alat za web aplikacije

❑ Dodavanje novih grupa korisnika

- Enable Roles → Create or Manage Roles

The screenshot shows the Microsoft ASP.NET Web Site Administration Tool interface. At the top, there's a header with the ASP.NET logo and the title "Web Site Administration Tool". A navigation bar contains four tabs: "Home", "Security" (which is selected), "Application", and "Provider". Below the navigation bar, there's a text block explaining that roles can be added to control access to specific folders. Below this, there's a section titled "Create New Role" with a text input field for "New role name:" and an "Add Role" button. At the bottom, there's a table with two columns: "Role Name" and "Add/Remove Users". The table contains one row for the role "Dobavljac" with links for "Manage" and "Delete".

ASP.NET Web Site Administration Tool [How do I use this tool?](#)

Home **Security** Application Provider

You can optionally add roles, or groups, that enable you to allow or deny groups of users access to specific folders in your Web site. For example, you might create roles such as "managers," "sales," or "members," each with different access to specific folders.

Create New Role

New role name:

Role Name	Add/Remove Users
Dobavljac	Manage Delete

Ugrađeni administracijski alat za web aplikacije

❑ Dodavanje novih korisnika

The screenshot shows the 'ASP.NET Web Site Administration Tool' interface. At the top, there's a navigation bar with 'Home', 'Security' (selected), 'Application', and 'Provider' tabs. Below the navigation bar, a message says: 'Add a user by entering the user's ID, password, and e-mail address on this page.'

The main content area is divided into two panels. The left panel, titled 'Create User', contains the 'Sign Up for Your New Account' form. The right panel, titled 'Roles', shows the role selection options.

Create User Panel:

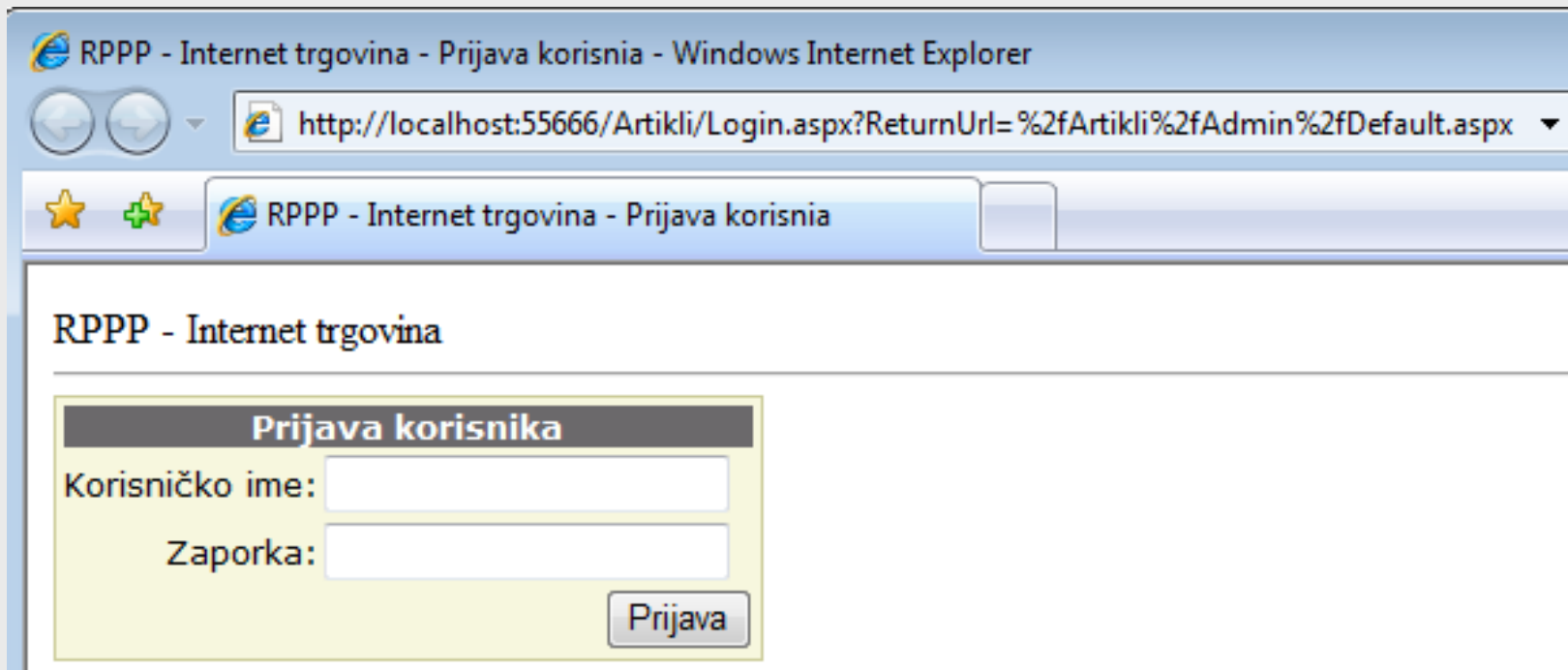
- Sign Up for Your New Account**
- User Name:
- Password:
- Confirm Password:
- E-mail:
- Security Question:
- Security Answer:
-

Roles Panel:

- Select roles for this user:
- ☒ Dobavljači

Primjer provjere autentičnosti kroz formu

- ❑ Nakon što korisnik pokuša otići na stranicu `http://localhost/Artikli/Admin/Default.aspx` pojavljuje se ekran za prijavu korisnika



The screenshot shows a Windows Internet Explorer browser window. The title bar reads "RPPP - Internet trgovina - Prijava korisnia". The address bar shows the URL "http://localhost:55666/Artikli/Login.aspx?ReturnUrl= %2fArtikli%2fAdmin%2fDefault.aspx". The browser's Favorites bar shows a single item: "RPPP - Internet trgovina - Prijava korisnia". The main content area displays the text "RPPP - Internet trgovina" followed by a login form. The form has a title "Prijava korisnika" in a dark header. Below the header are two input fields: "Korisničko ime:" and "Zaporka:". A "Prijava" button is located at the bottom right of the form.

- ❑ Nakon uspješne prijave, dolazi do automatskog preusmjeravanja na prethodno zahtjevanu stranicu

Ostalo

Praćenje rada Web aplikacije

❑ **System.Web.TraceContext**

- postupci `Trace.Write`, `Trace.Warn` (crveno)
- svojstva `Trace.IsEnabled`, `Trace.Mode` (`SortByTime`, `SortByCategory`)

❑ **Trag (trace)**

- na razini stranice

```
<%@ Page Trace="True" %>
```

```
...
```

```
Trace.Write("MyApp", "Button Clicked");
```

```
Trace.Write("MyApp", "Value: " + value);
```

- na razini aplikacije, `Web.config`

```
<configuration>
```

```
  <system.web>
```

```
    <trace enabled="true" pageOutput="true"  
      requestlimit="10"/>
```

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

```
</configuration>
```

❑ **Primjer: Web\PracenjeRada.aspx**

Objekti ASP.NET aplikacije

- ❑ **Svaka Web aplikacija nasljeđuje `System.Web.HttpApplication`**
- ❑ **Neki članovi `System.Web.HttpApplication`**
 - `Application` - stanje Web aplikacije
 - `Request` – HTTP zahtjev za stranicom
 - `Response` – odgovor na HTTP zahtjev
 - `Server` – računalo poslužitelj
 - `Session` – posjet Web aplikaciji od strane istog korisnika
 - `User` – podaci o korisniku
- ❑ **Važniji događaji koji se mogu obraditi u `Global.asax`:**
 - `Application_Start` – kada se dohvati prva stranica Web aplikacije
 - `Application_End` – kada se Web server zaustavi
 - `Session_OnStart` – kada korisnik prvi puta pristupi Web aplikaciji
 - `Session_End` – korisnik se odjavio ili je isteklo dodijeljeno mu vrijeme

Upravljanje stanjem aplikacije

❑ Stanje se pohranjuje u instancu `HttpApplicationState`

- može se dohvatiti putem svojstva `Page.Application`
- moguće zaključavanje `Application` objekta pri konkurentnom pristupu

❑ Sjednica (`session`)

- kontekst uporabe iste web aplikacije
- identifikator: 120 bit ASCII string
- varijable: pohrana podataka između opetovanih pristupa stranici
- parovi (ključ, vrijednost)
- Postavljanje vrijednosti za neki ključ
`Session["kljuc"] = vrijednost`
- Dohvat vrijednosti za neki ključ. Vraća `null` ako ne postoji vrijednost za zadani ključ ili vrijednost tipa `object` (potreban `cast`).

`Session["kljuc"]`

- Brisanje nekog ključa iz trenutne sjednice

`Session.Remove("naziv")`

- Brisanje svih ključeva iz trenutne sjednice

`Session.RemoveAll()`

Sjednica

❑ Standardno, session identifikator pohranjuje se u cookie

- može se onemogućiti u `Web.config`

```
<configuration>
  <system.web>
    <sessionState cookieless="true" timeout="5"/>
  </system.web>
</configuration>
```

- U tom slučaju se identifikator prenosi kao parametar unutar URL-a

❑ Primjer: `Web\PracenjeRada.aspx`

Prijenos argumenata stranici

❑ Standardno URL je oblika

`http://server/putanja/stranica.aspx?naziv1=vrijednost1&naziv2=vrijednost2`

❑ Vrijednost pojedinog parametra iz URL-a može se dobiti pomoću

- `Request["Naziv parametra"]`

❑ Neka od svojstava klase `Request`:

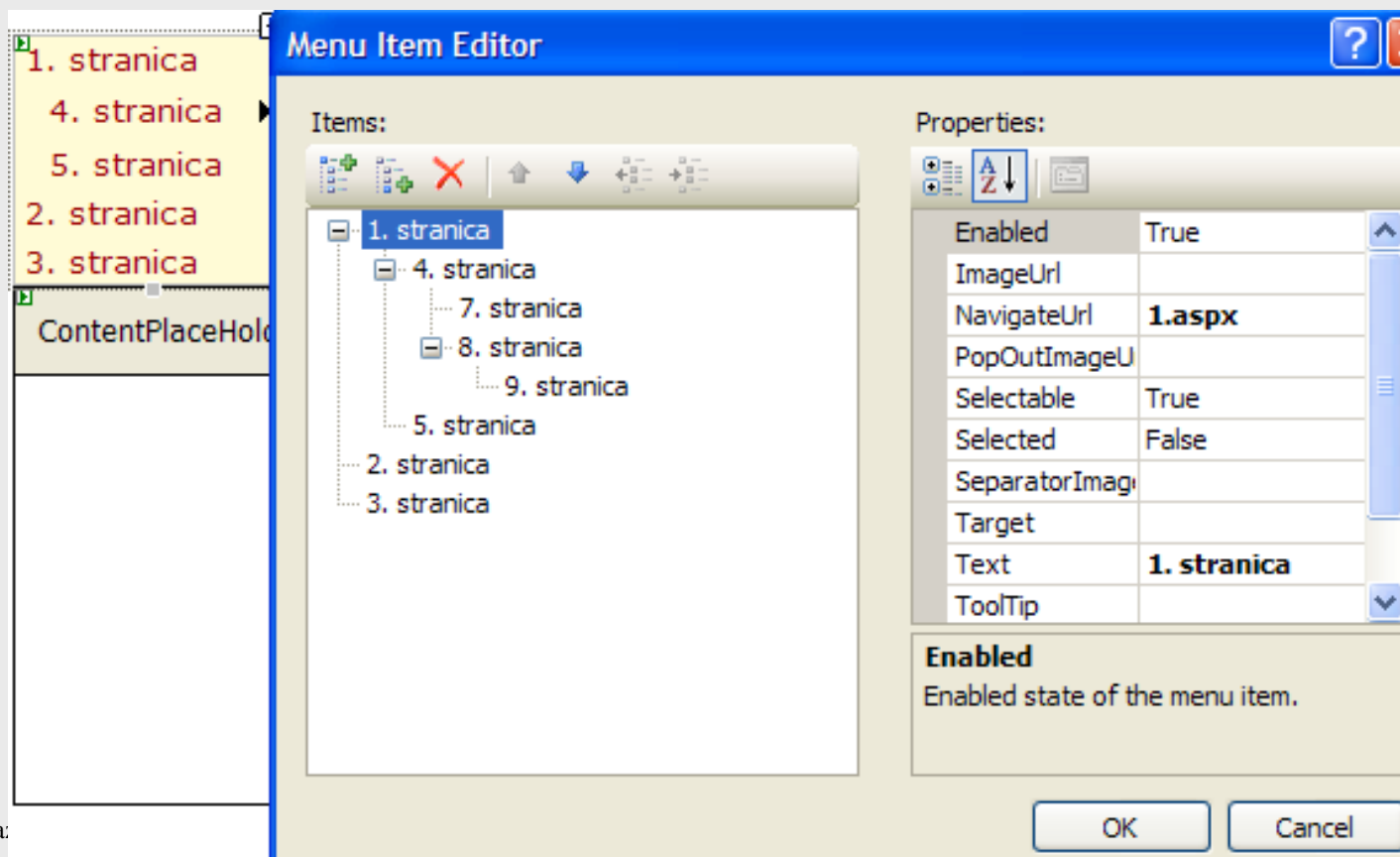
- `UserHostAddress` : IP adresa klijenta (posjetitelja stranice)
- `UserAgent` : Internet preglednik klijenta
- `Request.Url.OriginalString`: Puni URL trenutne stranice
- `Request.Url.LocalPath` : Lokacija trenutne stranice na serveru (kazalo u kojem se aplikacija nalazi)

Izbornici

- ❑ **Svaki izbornik je poveznica na određenu stranicu**
- ❑ **Dvije vrste izbornika:**
 - Prikaz izbornika slično kao kod Windows aplikacija (dinamički se prikazuju iz korijena izbornika) – kontrola *Menu*
 - Prikaz u obliku stabla - kontrola *TreeView*
- ❑ **Elementi izbornika mogu se navoditi:**
 - direktno unutar kontrole za navigaciju
 - definiranjem izvora podataka koji se veže na određenu mapu web-sjedišta
- ❑ **Trenutni položaj određene stranice i veza prema nadređenim stranicama može se prikazati kontrolom *SiteMapPath***
 - Za korištenje kontrole *SiteMapPath* potrebno je imati definiran izvor podataka za navigaciju, odnosno mapu web sjedišta.
- ❑ **Mapa web sjedišta je datoteka s ekstenzijom *.sitemap***
 - *Add New Item -> Site Map*
 - Sadrži hijerarhijski popis stranica uključenih u izbornik navigacije
 - Pretpostavljeni (eng. default) naziv mape web mjesta je *web.sitemap*

Kontrola *Menu* – važnija svojstva

- ☐ **Orientation** – orijentacija prikaza prvog nivoa izbornika (horizontalno/vertikalno).
- ☐ **StaticDisplayLevels** – broj nivoa koji će inicijalno biti prikazani
- ☐ **Items** – definira elemente u izborniku
- ☐ **DataSourceID** – izvor podataka (ako se ne koristi svojstvo **Items**)



Primjer: Web\Izbornici – MasterPage.master, kontrola Menu, direktno navođenje elemenata izbornika


```
<asp:Menu ID="Menu1" runat="server" StaticDisplayLevels="2">
<Items>
    <asp:MenuItem NavigateUrl="1.aspx" Text="1. stranica"
        Value="1. stranica">
        <asp:MenuItem NavigateUrl="4.aspx" Text="4. stranica"
            Value="4. stranica">
            <asp:MenuItem NavigateUrl="7.aspx" Text="7. stranica"
                Value="7. stranica"></asp:MenuItem>
            <asp:MenuItem NavigateUrl="8.aspx" Text="8. stranica"
                Value="8. stranica">
                <asp:MenuItem NavigateUrl="9.aspx" Text="9.
stranica"
                    Value="9. stranica"></asp:MenuItem>
            </asp:MenuItem>
        </asp:MenuItem>
        <asp:MenuItem NavigateUrl="5.aspx" Text="5. stranica"
            Value="5. stranica"></asp:MenuItem>
    </asp:MenuItem>
    <asp:MenuItem NavigateUrl="2.aspx" Text="2. stranica"
        Value="2. stranica"></asp:MenuItem>
    <asp:MenuItem NavigateUrl="3.aspx" Text="3. stranica"
        Value="3. stranica"></asp:MenuItem>
</Items>
</asp:Menu>
```

Definiranje mape web sjedišta

❑ Primjer: Web\Izbornici – web.sitemap

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap
xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-
1.0" >
  <siteMapNode url="" title="Izbornik">
    <siteMapNode url="1.aspx" title="1. stranica" >
      <siteMapNode url="4.aspx" title="4. stranica" >
        <siteMapNode url="7.aspx" title="7. stranica" />
        <siteMapNode url="8.aspx" title="8. stranica" >
          <siteMapNode url="9.aspx" title="9. stranica"
/>
        </siteMapNode>
      </siteMapNode>
    <siteMapNode url="5.aspx" title="5. stranica" />
  </siteMapNode>
  <siteMapNode url="2.aspx" title="2. stranica" />
  <siteMapNode url="3.aspx" title="3. stranica" />
</siteMapNode>
</siteMap>
```

Povezivanje izbornika i mape web sjedišta

- ❑ Primjer:  Web\Izbornici – MasterPage.master
- ❑ Na stranici je potrebno definirati izvor podataka za mapu web sjedišta – ***SiteMapDataSource***
 - dodavanjem na stranicu ili, primjerice, klikom na kontrolu nekog od izbornika, a zatim *Choose Data Source -> New Data Source -> Site Map*
 - ako se koriste izbornici iz datoteke web.sitemap nije potrebno dodatno podešavanje parametara izvora podataka
- ❑ Kontroli ***Menu*** ili ***TreeView*** postaviti ***DataSourceID*** na naziv prethodno dodanog izvora podataka

```
<asp:TreeView ID="TreeView1" runat="server"
    DataSourceID="SiteMapDataSource1" >
    ...
</asp:TreeView>
<asp:SiteMapDataSource ID="SiteMapDataSource1"
runat="server" />
```


Definiranje više mapa unutar web sjedišta

- ❑ Web sjedište može imati više mapa
- ❑ Nije nužno imati mapu *web.sitemap*
- ❑ U datoteci *web.config* mogu se definirati parametri za mape

```
<system.web>
  <siteMap defaultProvider="GlavniIzbornik">
    <providers>
      <add name="GlavniIzbornik"
            type="System.Web.XmlSiteMapProvider"
            siteMapFile="~/Navigacija/glavni.sitemap"/>
      <add name="Izbornik2"
            type="System.Web.XmlSiteMapProvider"
            siteMapFile="~/Navigacija/izbornik2.sitemap"/>
    </providers>
  </siteMap>
  ...
```

- ❑ Prilikom definiranja izvora podataka za mapu potrebno je navesti pružatelja usluge za mapu web sjedišta

```
<asp:SiteMapDataSource ID="Izbornik2SiteMapDataSource"
  runat="server"
  SiteMapProvider="Izbornik2" />
```

WCF servisi (osnove)

11/13

WCF servisi


❑ Neki nedostatci klasičnog web servisa:

- Razredi definirani na strani web servisa
- Generirani razredi sadrže samo javna svojstva i javne varijable
- Postupci se ne serijaliziraju
- Razredi koji nasljeđuju `IDictionary` ne mogu se serijalizirati (pr. `Hashtable`)

❑ WCF


- Ne mora nužno biti web servis (HTTP, TCP, Net Pipes...)
- ABC princip : *Address, Binding, Contract*
 - Adresa i način povezivanja:
 - Zapisano u datotekama *web.config/app.config*
 - Ugovor: skup sučelja, razreda i postupaka
 - sadrže ga obje strane u komunikaciji
 - neovisan o adresi i načinu povezivanja
- Jasnije odvajanje razreda za komunikaciju
- Jednostavnija promjena razreda, implementacija i smještaja

Definiranje sučelja s postupcima u ugovoru

- ❑ **Primjer:**  **WCF\Ugovor\IFirmaServis.cs**
 - Opisuje postupke koji će se koristiti u komunikaciji
 - Ne određuje vrstu komunikacije i način implementacije
- ❑ **Prostor imena `System.ServiceModel`**
 - Add Reference -> **`System.ServiceModel.dll`**
- ❑ **Sučelje dekorirano atributom `[ServiceContract]`**
- ❑ **Postupak dekoriran atributom `[OperationContract]`**

```
[ServiceContract]
public interface IFirmaServis
{
    [OperationContract]
    List<Drzava> PopisDrzava();
}
```

Razredi u komunikaciji s WCF servisom

- ❑ Primjer:  WCF\Ugovor\Drzava.cs
- ❑ Prostor imena `System.Runtime.Serialization`
 - Add Reference -> `System.Runtime.Serialization.dll`
- ❑ Razred dekoriran atributom `[DataContract]`
- ❑ Svojstva dekorirana atributom `[DataMember]`

```
[DataContract]
```

```
public class Drzava  
{
```

```
    [DataMember]
```

```
    public string Oznaka { get; set; }
```

```
    [DataMember]
```

```
    public string Naziv { get; set; }
```

```
    [DataMember]
```

```
    public string Iso3Oznaka { get; set; }
```

```
    public override string ToString()
```

```
    {
```

```
        return string.Format("{0}: {1}, {2}", Oznaka...
```

```
    }
```

```
}
```

Implementacija sučelja

❑ **Primjer:**  **WCF\ImplementacijaUgovora\FirmaServis.cs**

❑ **Zaseban projekt**

- Referencirati ugovor
- Implementirati sučelja iz ugovora

```
public class FirmaServis : Ugovor.IFirmaServis
{
    public List<Ugovor.Drzava> PopisDrzava()
    {
        ...
    }
}
```

Smještaj WCF Servisa na web mjestu

- ❑ Dodati referencu na projekt koji sadrži implementaciju ugovora
- ❑ Napraviti datoteku `FirmaServis.svc` sa sadržajem:

```
<%@ ServiceHost Service="ImplementacijaUgovora.FirmaServis" %>
```

- ❑ U datoteci `web.config` definirati pristupnu točku (endpoint) za konkretnu implementaciju, definirati način povezivanja (binding) i specificirati o kojem se ugovoru radi (contract)

```
<system.serviceModel>
  <behaviors><serviceBehaviors>
    <behavior name="FirmaServisBehavior">
      <serviceMetadata httpGetEnabled="true" />
      <serviceDebug includeExceptionDetailInFaults="true" />
    </behavior></serviceBehaviors></behaviors>
  <services>
    <service behaviorConfiguration="FirmaServisBehavior"
      name="ImplementacijaUgovora.FirmaServis">
      <endpoint address="" binding="wsHttpBinding"
        contract="Ugovor.IFirmaServis"/>
    </service>
  </services>
</system.serviceModel>
```

Poziv WCF servisa kao poziv web servisa

❑ Poziv kao da se radi o klasičnom web servisu

❑ Primjer:  WCF\PozivKaoWebServis

- Dodati referencu (Add Service Reference ...)
- Automatski generiran klijent za poziv web servisa, a adresa zapisana u `web.config`

```
FirmaWebServis.FirmaServisClient client = new  
FirmaWebServis.FirmaServisClient();  
IspisiDrzave(client.PopisDrzava());
```


Poziv WCF servisa

❑ Poziv WCF servisa (Primjer: WCF\PozivKaoWCFServis)

- Referencirati ugovor
- Klijent se stvara za pristupnu točku definiranu u datoteci web.config
- U datoteci **app.config** se navodi način komunikacije

```
string EndPointName = "FirmaEndPointWebServis";  
//string EndPointName = "FirmaEndPointLokalno";  
using (ChannelFactory<Ugovor.IFirmaServis> factory =  
    new ChannelFactory<Ugovor.IFirmaServis>(EndPointName))  
{  
    Ugovor.IFirmaServis client =  
        factory.CreateChannel();  
    IspisiDrzave(client.PopisDrzava());  
}
```

- Razred `Drzava` definiran u ugovoru može imati i postupke
 - U ovom primjeru je to prekrivanje postupka **Tostring()**
 - Zajednička funkcionalnost dostupna i klijentu i serveru napisana na jednom mjestu


Definicija pristupne točke

- ❑ **Pristupna točka sadrži** ( **WCF\PozivKaoWCFServis\app.config**)
- Naziv razreda iz ugovora
 - Naziv pristupne točke
 - Adresu servisa
 - Način povezivanja (wsHttpBiding, netNamePipeBinding...)

```
<system.serviceModel>
  <client>
    <endpoint contract="Ugovor.IFirmaServis"
              name="FirmaEndPointWebServis"
address="http://localhost:29076/SmjestajWeb/FirmaServis.svc"
              binding="wsHttpBinding"      />

    <endpoint contract="Ugovor.IFirmaServis"
              name="FirmaEndPointLokalno"
address="net.pipe://localhost/FirmaServis"
              binding="netNamedPipeBinding" />
  </client>
</system.serviceModel>
```

Povezivanje WCF servisa korištenjem cjevovoda

- ❑ Smještaj implementacije ugovora unutar windows aplikacije
- ❑ Komunikacija korištenjem cjevovoda
- ❑ Primjer:  WCF\SmjestajWinApp

```
ServiceHost host =  
    new ServiceHost(typeof(ImplementacijaUgovora.FirmaServis),  
        new Uri[] { new Uri("net.pipe://localhost") });  
  
NetNamedPipeBinding binding = new NetNamedPipeBinding();  
  
host.AddServiceEndpoint(typeof(Ugovor.IFirmaServis),  
    binding, "FirmaServis");  
  
host.Open();
```

- ❑ Na klijentu je potrebno samo promijeniti parametre pristupne točke



Zadaci za vježbu

- ☐ Proširiti web servis za podatke o prodaji tako da se uzimaju narudžbe od nekog određenog datuma.
- ☐ Dovršiti stranicu `Admin/Default.aspx` na način da se kreiraju kontrole koje omogućuju unos željenog datumskog intervala za koji će se dobiti podaci o prodaji.
- ☐ Proučiti kontrolu *FileUpload* i napisati web aplikaciju koja će omogućiti slanje e-maila s pravitkom koji će korisnik prethodno poslati na server.

Reference

- <http://www.asp.net/>
- <http://msdn2.microsoft.com/en-us/asp.net/default.aspx>
- <http://www.asp.net/ajax/>
- <http://msdn2.microsoft.com/en-us/library/aa480453.aspx>
- <http://www.w3schools.com/html/>
- <http://www.w3schools.com/xml/>
- <http://www.dotnetnuke.com/>
- <http://www.developerfusion.co.uk/show/25/20/> ili
<http://www.sitepoint.com/article/asp-net-server-controls>
- <http://www.15seconds.com/issue/050216.htm>
- <http://msdn.microsoft.com/en-us/netframework/aa663324.aspx>
- <http://msdn.microsoft.com/en-us/library/ms730214.aspx>