

Serverske tehnologije

IIS, ASP.NET

Ak. god. 2010/2011

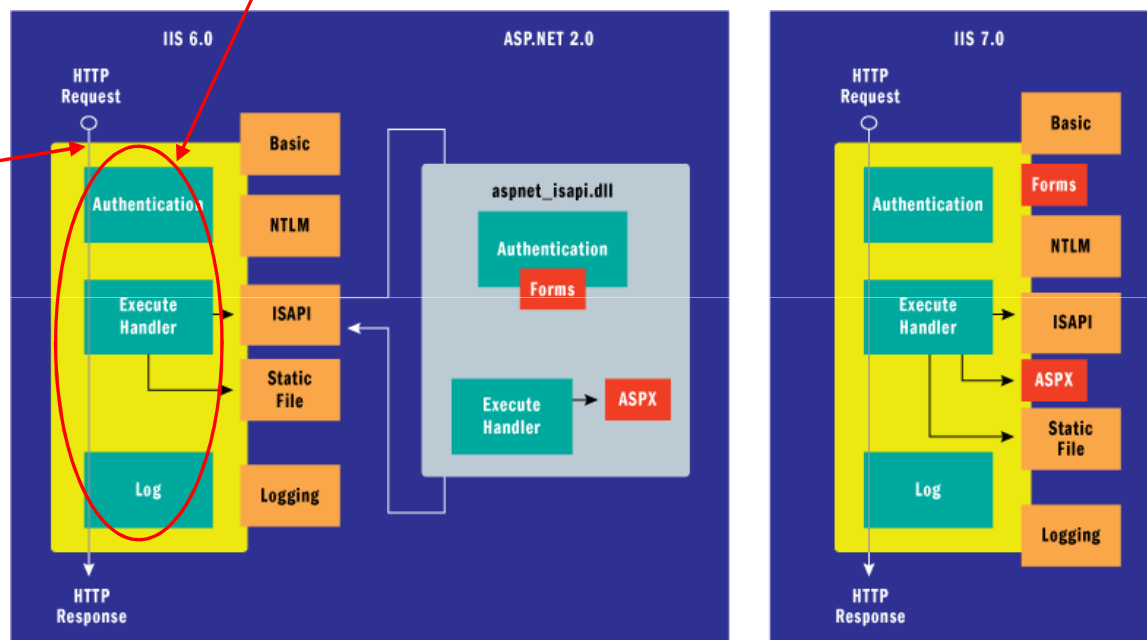
IIS

- IIS (*Internet Information Server*) je Microsoftova aplikacija koja u sebi uključuje HTTP server (pored FTP, SMTP i NNTP servera).
- Kada IIS servis vraća zahtijevani URI, može klijentu odgovoriti na tri načina:
 - Može vratiti statički dokument – Statička datoteka sa nalazi na disku i za njeno dohvaćanje IIS koristi usluge File Systema servera. Nakon dobivanja zahtjeva IIS provjerava korisnička prava i vraća korisniku zahtijevani dokument.
 - Može izvršiti CGI skriptu – Kada klijent pošalje zahtjev za izvršnom CGI skriptom IIS pokreće izvršavanje skripte i vraća klijentu rezultat izvršavanja.
 - Može pokrenuti neki od instaliranih ISAPI proširenja (*Internet Server Application Programming Interface*) proširenja.
- Glavna prednost ISAPI tehnologije u odnosu na CGI je što IIS ne treba pokretati novi zasebni proces pa je ova tehnologija brža i manje opterećuje procesor. ISAPI koristi DLL funkcijske pozive za komunikaciju sa komponentama za proširenje, a ne standardni I/O i okolinske varijable kao što to radi CGI.

IIS

Integrirani moduli

- IIS 7 preporuča razvoja ekstenzija korištenjem modularnog API-ja umjesto ISAPI-ja. Modularni API omogućava da se modul uključi bilo gdje u cjevovodu (*pipeline*) obrađivanja HTTP zahtjeva.
- Moduli mogu biti globalni i koriste se na svim site-ovima na serveru ili lokalni i ograničeni samo na određeni site.
- IIS 7 podržava tzv. integrirani *application pool* način kada se ASP.NET moduli uključuju u cjevovodu kao modularni API dok klasični *application pool* način uključuje ASP.NET ako ISAPI.



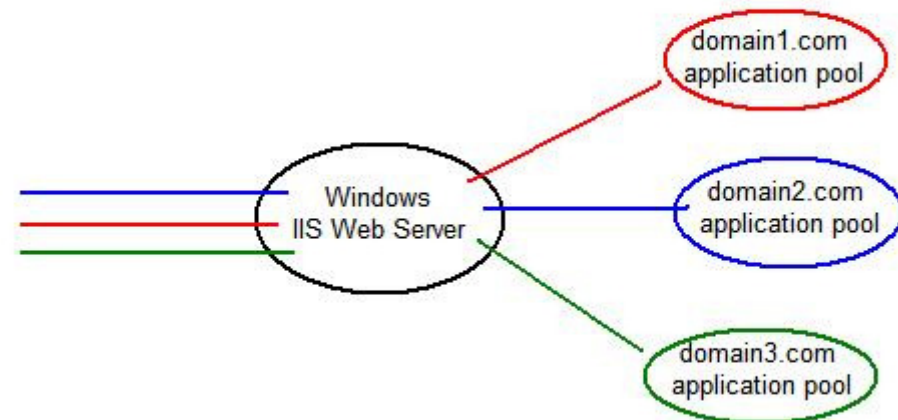
NTLM (NT LAN Manager) je Microsoftov autentikacijski protokol koji koristi *challenge-response* sekvencu u kojoj klijent koji se hoće autentificirati i server koji traži autentikaciju klijenta razmjene tri poruke.

IIS

- Windows IIS web server implementira koncept koji se zove *application pool* i koji je napravljen kako bi omogućio izolaciju procesa koji se izvode na serveru u svrhu stabilnosti i sigurnosti.
- Svaki *application pool* može sadržavati jednu ili više aplikacija (praktički svaka asp.net skripta je aplikacija). Kada se aplikacije izvode u zasebnim *application pool-ovima* greške u jednoj aplikaciji ne utiču na aplikacije koje se izvode u drugim *application pool-ovima*. *Application pool* može prilagođavati stupanj izolacije koji osigurava aplikaciji.
- Kada se doda novi web site on se automatski dodaje u *Default Application Pool* (DefaultAppPool). Ukoliko se ne kreira novi *application pool* i poveže sa web site-om svi web site-ovi se izvode u istom *application pool-u*.

IIS

- Dobro je da se svaka domena (na istom IIS-u, može se *hostati* više različitih domena) izvodi u vlastitom *application pool-u*.
- IIS statičke dokumente isporučuju klijentu bez obrade.
- Ali bilo koji dinamički dokument poput ASP, ASP.NET ili nekog drugog tipa aplikacije se mora izvršiti. IIS predaje taj HTTP zahtjev u *application pool-u* koji je zasebni proces i unutar njega se izvršava dinamički dokument bez utjecaja na druge *application pool-ove*.



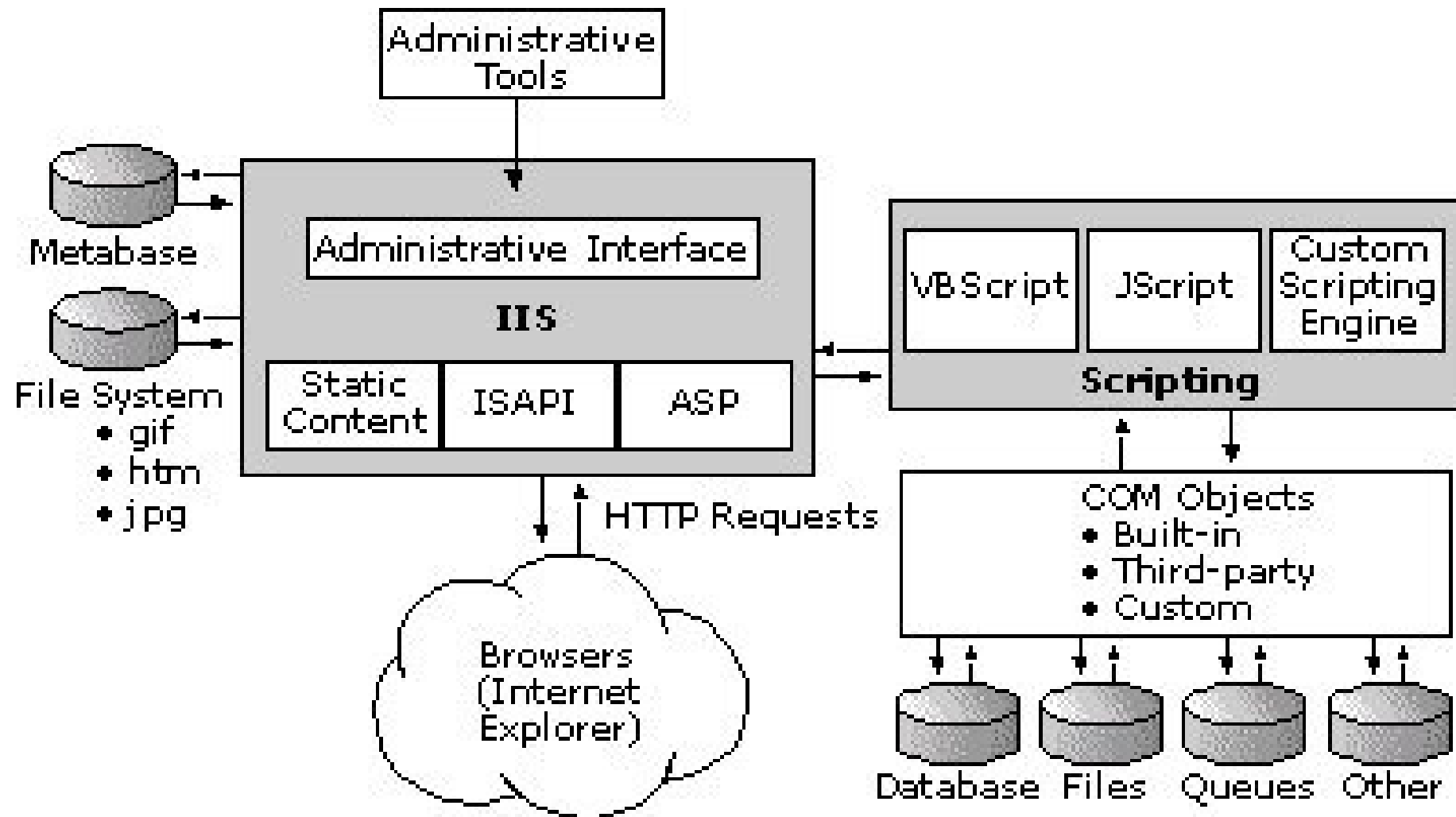
IIS

- Kod “starijih” verzija IIS (prije verzije 6.0) konfiguracijske informacije servera se pohranjuju u IIS metabazi (Metabase), sličnoj registru Windowsa po hijerarhijskoj strukturi.
- Od verzije IIS-a 6.0 koja se vrti na Windows Severu 2003, Windows XP i Visti konfiguracijske informacije servera se pohranjuju u XML formatu.
- Zadnja verzija IIS je 7.0 i može se instalirati na Visti i Serveru 2008.
- Administriranje konfiguracije IIS servera se radi preko aplikacije MMC (*Microsoft Management Console*).
(C:\WINDOWS\system32\inetsrv\InetMgr.exe)

IIS

- IIS koristi virtualne direktorije (*virtual directories*) za referenciranje datoteka umjesto punog imena datoteka (*fully qualified path names*).
- Virtualni direktorij je drugo ime (*alias*) za stvarni direktorij. Na taj način je moguće distribuirati datoteke IIS-a na različita mjesta na računalu, pa čak i na različita Windows računala spojena na mrežu.
- Početni direktorij (*home directory*) za IIS je root stabla virtualnog direktorija. Drugo ime se povezuje sa virtualnim direktorijom kada je direktorij dodan u IIS konfiguraciju (najjednostavnije preko MMC-a). Drugo ime je ime koje klijenti koriste kao ime poddirektorija Web servera kada žele pristupiti datotekama u tom direktoriju.

IIS arhitektura



ASP.NET



- ASP.NET omogućava izvršavanje skriptnog koda na strani poslužitelja. To je novija verzija ASP – *Active Server Pages* tehnologije bazirana na .NET platformi. Programski kod se izvršava unutar IIS-a.
- ASP.NET podrazumijeva centralizirane aplikacije smještene na jednom ili više poslužitelja, koje dinamički odgovaraju na zahtjeve klijenta.
- Odgovori su dinamički jer ASP.NET presreće zahtjeve stranica prema određenim ekstenzijama, na primjer aspx ili ascx i preuzima odgovornost da odgovara na te zahtjeve datotekama just-in-time prevedenog koda.

ASP.NET



- ASP.NET, za razliku od klasičnog ASP, omogućava da se cijeli kod aplikacije kompajlira u jedan ili više dll-ova na web serveru. Kompajliranje se radi automatski kada se stranici pristupi prvi put pa programer to ne mora raditi sam za svaku stranicu.
- ASPX dokumenti (ekstenzija ASP.NET stranica) i resurs datoteke su smještene u virtualnom direktoriju IIS-a. Kada klijent prvi put zatraži stranicu .NET framework iskompajlira sve što treba (stranicu, resurse, ..) u .NET assembly i pošalje odgovor. Budući zahtjevi se generiraju iz stvorenog DLL file-a. Po defaultu ASP.NET modul će iskompjalirati cijeli site u grupama od po 1000 datoteka kada dobije prvi HTTP zahtjev. Kako to ipak traje neko vrijeme, programer može sam iskopaljirati aplikaciju i postaviti već iskompajliranu aplikaciju na server.

ASP.NET



- ASP.NET stranica ili “web forma” je osnovni građevni blok aplikacije.
- Web forme se smještaju u datoteke sa ASPX ekstenzijom.
- Sadrže statički dio u obliku HTML ili XHTML, ali sadrže i markup oznake koje definiraju serverske (server-side) Web kontrole (Web Controls) i Korisničke kontrole (User Controls) u koje programer smješta sadržaj stranice.
- Dinamički kôd koji se izvršava na serveru može se na stranici smještati unutar bloka sa oznakama `<% kod %>` slično sintaksi PHP-a, (klasičnog) ASP-a, JSP-a, ali ovakav umetnuti kod (inline code) se ne preporuča, osim kod povezivanja sa izvorima podataka, jer uzrokuje više zahtjeva prema serveru prilikom iscrtavanja stranice.

ASP.NET



```
<%@ Page Language="C#" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<script runat="server">
    protected void Page_Load(object sender, EventArgs e)
    {
        Label1.Text = DateTime.Now.ToLongDateString();
    }
</script>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
    <title>Sample page</title>
</head>
<body>
    <asp:Label runat="server" id="Label1" />
</body>
</html>
```

- http://localhost/inline_primjer.aspx

ASP.NET



- Microsoft preporuča korištenje modela pozadinskog koda (code-behind model), pri čemu se kod odvaja u posebnu datoteku ili u zasebni imenovani skriptni tag.
- Datoteke pozadinskog koda obično se zovu isto kao i ASP.NET datoteka + ekstenzija ovisna o programskom jeziku (*MyPage.aspx.cs*, *MyPage.aspx.vb*), po ovom principu se automatski generira ime datoteka u Visual Studiju ili nekim drugim IDE-ovima. Programer u datotekama koda piše kod koji se izvodi u određenim događajima (event) kao što je učitavanje stranice, klikanje na botun i sl..
- ASP.NET model pozadinskog koda omogućava odvajanje prezentacijskog dijela aplikacije od sadržaja aplikacije čime se olakšava razvoj i održavanje aplikacije slično kao kod MVC (model-view-controller) predloška.

ASP.NET



```
<%@ Page Language="C#" CodeFile="code_behinde_primjer.aspx.cs"
    Inherits="Website.code_behinde_primjer" AutoEventWireup="true" %>
<!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>Sample page</title>
</head>
<body>
    <asp:Label runat="server" id="Label1" />
</body>
</html>
```

- Direktiva `<%@ Page Language="C#" CodeFile="code_behinde_primjer.aspx.cs" Inherits="Website.code_behinde_primjer" AutoEventWireup="true" %>` se smješta na početku ASPX datoteke. Svojstvo *CodeFile* u naredbi *@ Page* specificira .cs ili .vb datoteku pozadinskog koda dok svojstvo *Inherits* specificira klasu.

ASP.NET

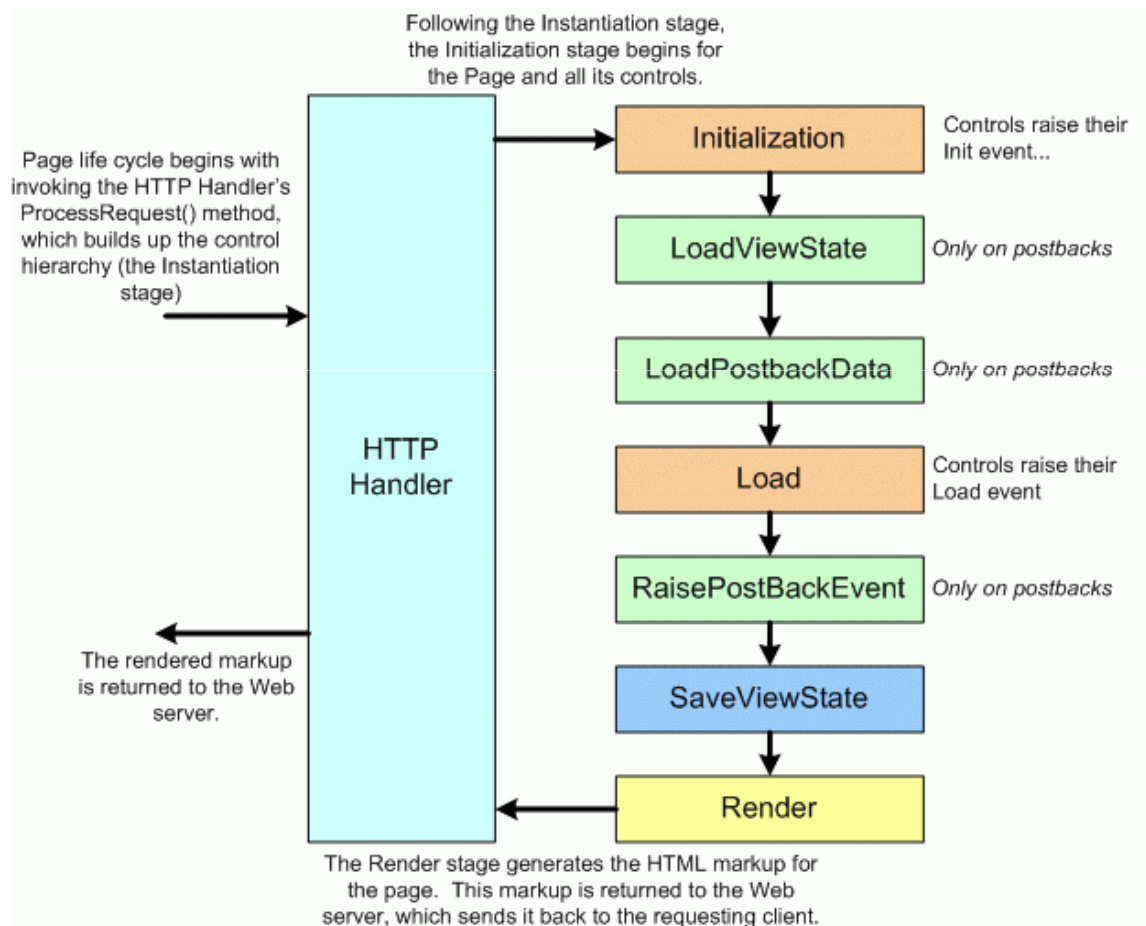
```
code_behinde_primjer.aspx.cs
using System;
namespace Website
{
    public partial class code_behinde_primjer : System.Web.UI.Page
    {
        // ova radi na AutoEventWireup="true"
        protected void Page_Load( object sender, System.EventArgs e )
        {
            Label1.Text = DateTime.Now.ToLongDateString();
            Label2.Text = "PAGE LOAD";
        }
        // ova radi na AutoEventWireup="false"
        protected override void OnLoad(EventArgs e)
        {
            Label1.Text = DateTime.Now.ToLongDateString();
            Label2.Text = "ONLOAD";
            base.OnLoad(e);
        }
    }
}
```

ASP.NET



- Ako je svojstvo `AutoEventWireup` postavljenu u `true` (ili ako nedostaje, defaultna vrijednost je `true`), događaji na stranici se pozivaju automatski osobito `Page_Init` i `Page_Load` metoda.
- Nije potrebno posebno definirati event handler za određeni event.
- Nedostatak je da metode za obrađivanje eventa trebaju imati točno određeno ime, dok ukoliko se metoda definira preko event handlera možemo staviti kakvo želimo ime metode.
- `OnLoad` je ekvivalentna metoda u C# koja se poziva kada je svojstvo `AutoEventWireup` postavljeno u `false`.
- U primjeru će se `Page_Load()` metoda ili `OnLoad()` metoda pozvati svaki put kada se stranica učitava. Obrada događaja može se povezati uz različite faze izvršavanja stranice.

ASP.NET



Životni ciklus
ASP.NET stranice

ASP.NET



- U svakoj fazi izvršavanja stranice, stranica podiže određene događaje koji se mogu obraditi u kodu.
- Za povezivanje na događaje koje podižu kontrole na stranici (botun, ...) uz događaj se povezuje event handler ili u kôdu ili u ASP.NET stranici koristeći deklarativne attribute tj. HTML attribute kontrole. Npr. kontrola button ima attribute onclick.
- Detaljnije o životnom ciklusu ASP.NET aplikacije
<http://msdn.microsoft.com/en-us/library/ms178472.aspx>

ASP.NET



- ASP.NET glavne stranice (master pages) omogućavaju kreiranje prikaza (master page) koji se može koristiti unutar više stranica sadržaja (content pages) koje zahtijevaju konzistentni prikaz. Održavanje prikaza jedne (ili nekoliko) glavne stranice olakšava održavanje izgleda cijelog site-a.
- Glavna stranica se sastoji od dva dijela, same glavne stranice i jedne ili više stranica sadržaja.
- Glavna stranica je ASP.NET datoteka sa ekstenzijom .master i definiranim prikazom koji može sadržavati statički tekst, HTML elemente i serverske kontrole. Glavna stranica je identificirana posebnom ASP.NET direktivom @ Master koja zamjenjuje direktivu @ Page kod običnih aspx stranica.
- `<%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>`

ASP.NET



- Pored @ Master direktive, glavna stranica sadrži i sve top-level HTML elemente stranice poput html, head, i form elementa.
- Npr. glavna stranica može sadržavati izbornik koji će se nalaziti na svim stranicama site-a, sliku koja je logo kompanije itd.
- Bilo koji HTML i ASP.NET element može se smjestiti na glavnu stranicu.
- Pored statičkog teksta i kontrola koje će se pojaviti na svim stranicama, glavna stranica sadrži i jednu ili više ContentPlaceHolder kontrola. Te kontrole definiraju gdje će se u glavnoj stranici smještati promjenjivi sadržaj iz drugih stranica koje koriste glavnu stranicu kao predložak.

```

<%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs"
    Inherits="MasterPage" %>
<!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 runat="server">
    <title>Untitled Page</title>
</head>
<body>
Izbornik <br />
<a href=Default.aspx>Prva stranica</a><br />
<a href=Default2.aspx>Druga stranica</a><br />
<a href=Default3.aspx>Treća stranica</a><br />
    <hr />
    <form id="form1" runat="server">
        <div>
            <asp:contentplaceholder id="ContentPlaceHolder1" runat="server">
                </asp:contentplaceholder>
            <asp:contentplaceholder id="ContentPlaceHolder2" runat="server">
                </asp:contentplaceholder>
        </div>
    </form>
</body>
</html>
E:\predavanja\internet_2\master_page

```

ASP.NET



- Povezivanje ContentPlaceHolder glavne stranice sa sadržajem pojedinačnih stranica sadržaja radi se na sljedeći način:
 - povezivanje stranice sa glavnim stranicom radi se preko direktive @ Page navođenjem atributa MasterPageFile koji kao vrijednost sadrži glavnu stranicu koja će se koristiti.

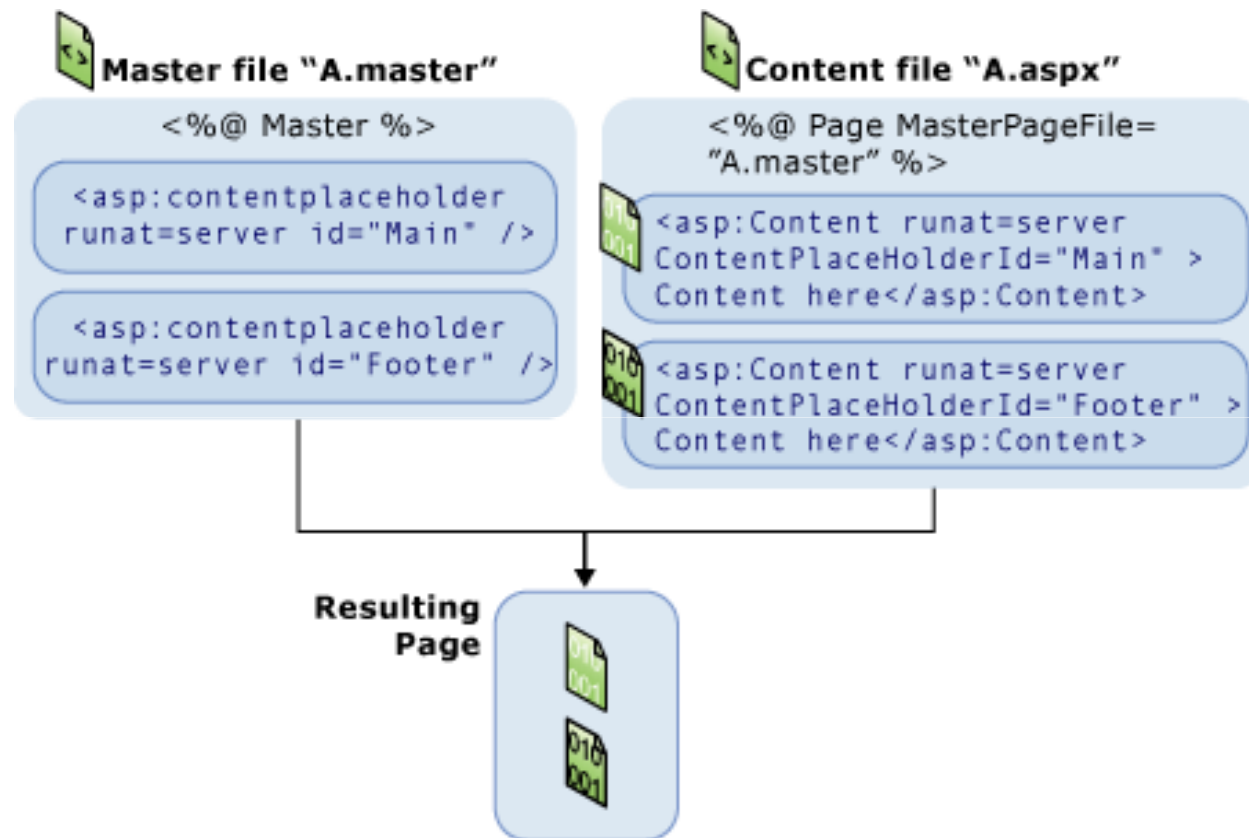
```
<%@ Page Language="C#" AutoEventWireup="true"  
    CodeFile="Default2.aspx.cs" Inherits="Default2"  
    MasterPageFile="~/MasterPage.master"%>
```

ASP.NET



- Sadržaj se kreira dodavanjem kontrola tipa Content i njihovim povezivanjem sa ContentPlaceHolder kontrolama na glavnoj stranici.

```
<%@ Page Language="C#" AutoEventWireup="true"
    CodeFile="Default3.aspx.cs" Inherits="Default3"
    MasterPageFile="~/MasterPage.master" %>
<asp:Content ID="Content1" runat="server"
    ContentPlaceHolderID="ContentPlaceHolder1">
    ovo je treća stranica</asp:Content>
<asp:Content ID="Content2" runat="server"
    ContentPlaceHolderID="ContentPlaceHolder2">
    ovo je treća stranica drugi placeholder</asp:Content>
```



ASP.NET



- Velika mana ASP-a je što se skriptni kôd morao ubaciti u HTML točno tamo gdje će se i prikazati. Složeniji projekti na taj način su često postajali teško razumljivi.
- ASP.NET taj problem rješava serverskim kontrolama, tagovima koje server može izvesti.
- Postoje više vrsta serverskih kontrola. To su:
 - HTML poslužiteljske kontrole – uobičajeni HTML tagovi
 - Web poslužiteljske kontrole – novi ASP.NET tagovi
 - Validacijske poslužiteljske kontrole – za validaciju unosa
 - Korisničke kontrole – vlastite kreirane kontrole

ASP.NET



- HTML elementi u ASP.NET datotekama su tretirani kao čisti tekst koji .NET framework ne obrađuje.
- HTML serverske kontrole su HTML elementi kojima je dodan atribut `runat="server"`. Taj atribut upućuje da se taj element tretira kao serverska kontrola, te je dostupna serverskim skriptama. Prilikom izvršavanja ASP.NET stranice se za svaki HTML element sa navedenim atributom kreira instanca odgovarajućeg objekta.
- Ukoliko je potrebno HTML element referencirati u kôdu elementu se pridodaje i atribut `id` koji kao vrijednost sadrži identifikator HTML elementa preko kojeg se elementu pristupa u kôdu.

- Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default"
    Debug="true"%>
```

```
<!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 runat="server">
```

```
    <title>Untitled Page</title>
```

```
</head>
```

```
<body>
```

```
    <form id="form1" runat="server">
```

```
        <div>
```

```
            &nbsp;<input id="Text1" type="text" runat="server" />
```

```
            <br /><br /><br />
```

```
            <input id="Button2" type="button" value="button" onclick="Button1_ServerClick" runat="server" />
```

```
            <br /><br /><br />
```

```
            <select id="Select1" style="width: 219px; height: 65px" runat="server">
```

```
                <option selected="selected"></option>
```

```
            </select>
```

```
            <br />
```

```
        </div>
```

```
    </form>
```

```
</body>
```

```
</html>
```

- E:\predavanja\internet_2\html_kontrole
- Atribut Debug omogućava debugiranje aplikacije.

- Default.aspx

```
using System.Web;
```

```
using System.Web.Security;
```

```
using System.Web.UI;
```

```
using System.Web.UI.WebControls;
```

```
using System.Web.UI.WebControls.WebParts;
```

```
using System.Web.UI.HtmlControls;
```

```
public partial class _Default : System.Web.UI.Page
```

```
{
```

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

```
    {
```

```
    }
```

```
    protected void Button1_ServerClick(object sender, EventArgs e)
```

```
    {
```

```
        ListItem dohvacen = Select1.Items.FindByValue(Text1.Value);
```

```
        if (dohvacen == null)
```

```
            Select1.Items.Add(Text1.Value);
```

```
    }
```

```
}
```

ASP.NET



- ASP.NET Web serverske kontrole su serverske kontrole specifične za ASP.NET.
- Također zahtijevaju `runat="server"` atribut.
- Razlika od HTML kontrole je u tome što Web serverska kontrola ne mora upućivati na neki konkretan HTML kôd, već može predstavljati mnogo kompleksniji element.
- Prilikom izvođenja ASP.NET stranice, Web serverska kontrola se iscrtava korištenjem odgovarajućih markup oznaka koje ovise ne samo o tipu browsera nego i postavkama kontrole. Npr. kontrola `TextBox` se može iscrtati kao HTML `input` tag ili kao HTML `textarea` tag ovisno o postavljenim svojstvima kontrole.
- `E:\predavanja\internet_2\web_kontrole`

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" Debug="true"%>
<!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 runat="server"><title>Untitled Page</title></head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:Table ID="Table1" runat="server" BorderStyle="Double" GridLines="Both" Height="168px"
        HorizontalAlign="Center" Width="320px">
        <asp:TableRow runat="server">
          <asp:TableCell runat="server">1,1</asp:TableCell>
          <asp:TableCell runat="server">1,2</asp:TableCell>
          <asp:TableCell runat="server">1,3</asp:TableCell>
        </asp:TableRow>
        <asp:TableRow runat="server">
          <asp:TableCell runat="server">2,1</asp:TableCell>
          <asp:TableCell runat="server">2,2</asp:TableCell>
          <asp:TableCell runat="server">2,3</asp:TableCell>
        </asp:TableRow>
        <asp:TableRow runat="server">
          <asp:TableCell runat="server">3,1</asp:TableCell>
          <asp:TableCell runat="server">3,2</asp:TableCell>
          <asp:TableCell runat="server">3,3</asp:TableCell>
        </asp:TableRow>
      </asp:Table>
      <br /><br />
      <asp:CheckBoxList ID="CheckBoxList1" runat="server" OnSelectedIndexChanged="CheckBoxList1_SelectedIndexChanged"
        Width="325px" AutoPostBack="True">
        <asp:ListItem>Prvi</asp:ListItem>
        <asp:ListItem>Drugi</asp:ListItem>
        <asp:ListItem>Treći</asp:ListItem>
      </asp:CheckBoxList>&nbsp;</div>
    </form>
  </body>
</html>

```

```

using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

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

    }

    protected void CheckBoxList1_SelectedIndexChanged(object sender, EventArgs e)
    {
        for (int i = 0; i < Table1.Rows.Count; i++)
            Table1.Rows[i].BackColor = System.Drawing.Color.White;

        Table1.Rows[CheckBoxList1.SelectedIndex].BackColor = System.Drawing.Color.Azure;
    }
}

```

- Po default-u, događaj `SelectedIndexChanged` neće odmah uzrokovati slanje zahtjeva prema serveru na obradu, a kod se izvršava na serveru. Događaj će se podignuti kada se forma sljedeći put pošalje serveru. Kako bi se odmah poslao zahtjev i podigao događaj potrebno je na kontroli postaviti `AutoPostBack` svojstvo u `true`.

ASP.NET



- ASP.NET Validacijske serverske kontrole se koriste pri provjeri unosa korisnika.
- Ako unos korisnika ne prođe validaciju, korisniku će se prikazati poruka pogreške.
- Svaka validacijska kontrola izvodi određenu vrstu validacije, kao što je provjera određene vrijednosti, ili raspona vrijednosti.
- Uglavnom, validacija se izvršava prilikom poziva nekog događaja (potvrda neke forme, klik na link itd.).

ASP.NET



- Ukoliko browser u kojem se učitava stranica podržava DHTML, ASP.NET validacijske kontrole mogu validaciju izvesti koristeći skripte na strani klijenta (JavaScript ili sl.).
- Kako za takvu validaciju nije potrebno slati zahtjev serveru brže je izvršavanje.
- Na kontrolama uglavnom nije potrebno ništa mijenjati kako bi koristile validaciju na strani klijenta jer kontrole automatski detektiraju DHTML podršku.
- Validacija na strani klijenta koristi isti mehanizam prikazivanja grešaka kao i validacija na strani klijenta.

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!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 runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:TextBox ID="TextBox1" runat="server" Width="155px"></asp:TextBox>
            <asp:Label ID="Label1" runat="server" Text="Ime"></asp:Label><br />
            <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ControlToValidate="TextBox1"
                ErrorMessage="Podatak ime je obavezan"></asp:RequiredFieldValidator><br />
            <asp:TextBox ID="TextBox2" runat="server" Width="156px"></asp:TextBox>&nbsp;
            <asp:Label ID="Label2" runat="server" Text="Prezime"></asp:Label><br />
            <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server" ControlToValidate="TextBox2"
                ErrorMessage="Podataka Prezime je obavezno"></asp:RequiredFieldValidator><br />
            <asp:TextBox ID="TextBox3" runat="server" Width="154px"></asp:TextBox>
            <asp:Label ID="Label3" runat="server" Text="Adresa"></asp:Label><br />
            <asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server" ErrorMessage="Podatak Adresa je
                obavezan" ControlToValidate="TextBox3"></asp:RequiredFieldValidator><br />
            <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Button" /></div>
        </form>
    </body>
</html>

```

E:\predavanja\internet_2\ validacijske_kontrole

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

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        // IsPostBack svojstvo je postavljenu u true kada je toPostBack zahtjev prema
        stranici, a ne prvi pristup stranici
        if (this.IsPostBack)
        {
            // Ukoliko validacija nije na strani klijenta onda je treba napraviti na
            strani servera
            this.Validate();
        }
    }
}
```