

ASP.NET = brže i lakše

Boris Bjelošević

U novembarskom broju našeg najomiljenijeg e-magazina smo pokazali kako se iz *Access*-ove baze **Northwind**, odnosno njene tabele **Customers**, mogu prikazati podaci na *web*-u. Da je kojim slučajem u prvoj liniji koda umesto

```
<% strProvider = "Driver={Microsoft Access Driver (*.mdb)};  
DBQ=C:\Inetpub\wwwroot\tutor\northwind.mdb"
```

stajalo

```
<% strProvider = "Driver={SQL Server}; Server=DADILJA; DataBase=Northwind; UID=sa;  
PWD=";
```

I kompletan ostatak koda ostao netaknut, na *web*-u bismo dobili podatke izvučene iz baze *Northwind*, tabele *Customers*, ali ovoga puta locirane na *SQL*-serveru koji se zove *Dadilja*. Sadržaj promenljive *strProvider* ne treba posebno objašnjavati. Valja samo napomenuti da, za razliku od prikazanog, prilikom instaliranja samog *SQL*-servera nije poželjno ostaviti korisničko ime administratora (username *sa*) bez šifre; izabranu šifru tada treba upisati odmah iza dela *PWD=* u promenljivoj *strProvider*.

Na manje ili više ozbiljnim *SQL*-serverima, gde se podaci unose na različite načine, obično se napravi jedan virtuelni korisnik sa svojom šifrom i svojim pravima pristupa (dodavanje, izmena i/ili brisanje podataka) koji se koristi samo za *ASP*-pristup bazi. U tom slučaju se ne koristi korisničko ime administratora baze, kao u gornjem primeru.

HTML-kód same stranice je napravio *ASP* na osnovu podataka dobijenih iz baze podataka i *html*-koda koji smo uneli ručno, a možemo ga videti u browser-u preko komande *View Source*.

Sada kad nam je poznat *web*-način prikazivanja podataka iz *Access*-a, odnosno *SQL Server*-a korišćenjem tehnologije *ASP*-a, pogledajmo kako se isti proces može uraditi korišćenjem *Microsoft*-ove tehnologije *.NET*.

Konkretan primer

Objašnjenja strukture i ideje samo *.NET*-a ćemo ostaviti za neki drugi broj ovog magazina. Ovde ćemo samo napomenuti da je za korišćenje prikazanih primera potrebno imati instaliran *.NET Framework* na *web*-serveru. Njegovu instalaciju možemo naći na *Microsoft*-ovom sajtu i „teška“ je oko 20MB.

Kód koji sledi ćemo detaljno objasniti, ali ga pre toga treba snimiti u fajl pod imenom *default.aspx*.

```
1. <%@ Import Namespace="System.Data" %>  
2. <%@ Import Namespace="System.Data.SqlClient" %>  
3. <html>  
4. <head>  
5. <title>ASP.NET u prkasi</title>  
6. <script language="VB" runat="server">  
7. Sub Page_Load(sender As Object, e As EventArgs)  
8. Dim myConn As SqlConnection  
9. Dim myComm As SqlDataAdapter  
10. Dim ds As DataSet  
11. myConn = New SqlConnection("server=DADILJA; DataBase=Northwind; UID=sa;  
PWD=";
```

```

12.     myComm = New SqlDataAdapter("SELECT * FROM Customers", myConn)
13.     ds      = New DataSet()
14.     myComm.Fill(ds)
15.     MyDataList.DataSource = ds
16.     MyDataList.DataBind()
17. End Sub
18.     </script>
19. </head>
20. <body>
21.     <asp:DataList id="MyDataList" RepeatColumns="2"
22.         RepeatDirection="Horizontal" runat="server">
23.         <ItemTemplate>
24.             <div style="padding:15,15,15,15;font-size:10pt;font-
family:Verdana">
25.                 <div style="font:12pt verdana;color:darkred">
26.                     <b><i><%# DataBinder.Eval(Container.DataItem,
"CompanyName") %></i></b>
27.                 </div>
28.                 <br>
29.                 <b>Customer ID:</b>
30.                 <%# DataBinder.Eval(Container.DataItem, "CustomerID") %><br>
31.                 <b>Contact Name:</b>
32.                 <%# DataBinder.Eval(Container.DataItem, "ContactName") %>, <%#
DataBinder.Eval(Container.DataItem, "ContactTitle") %><br>
33.                 <b>Address:</b>
34.                 <%# DataBinder.Eval(Container.DataItem, "Address") %><br>
35.                 <b>City:</b>
36.                 <%# DataBinder.Eval(Container.DataItem, "City") %>
37.             </div>
38.         </ItemTemplate>
39.     </asp:DataList>
40. </body>
41. </html>

```

I ovde smo koristili SQL-server pod imenom Dadija, bazu **Northwind** i tabelu **Customers**.

Prva dva reda našeg kóda omogućavaju da naš program koristi .NET-klase za rad sa bazama podataka (ove klase se nalaze u .NET Framework-u). Naredna tri reda su, nadamo se, već standardni i jasni za sve web-programere.

Šesti red naznačava da početak glavnog dela našeg programa. Deo runat mora da naglasi da se program izvrši na strani servera, te klijentska mašina dobija „gotov proizvod“. Deo language se u ovom slučaju može izostaviti, jer se za ASP.NET podrazumeva (default) jezik VB.NET; da smo kojim slučajem pisali program u jeziku C#, ovde bi obavezno morala da stoji naznaka da je korišćen taj jezik.

Naš program je prilično jednostavan i sadrži funkciju Page_Load, funkciju koja se izvršava čim se od servera zatraži prikazivanje samo stranice default.aspx. U njoj se odmah deklariraju lokalne promenljive: myConn (za konekciju na bazu konkretnog servera), myComm (za prosleđivanje sql-upita na bazom) i ds. Promenljiva ds ima ulogu da „preuzme“ podatke dobijene sql-upitom, što se postiže redom 14 našeg kóda. Ova promenljiva se donekle može zamisliti kao rs našeg ASP-primera, mada ima daleko više mogućnosti.

U redovima 15 i 16 sadržaj promenljive ds se prosleđuje ASP.NET kontroli DataList koju definišemo u 21. redu našeg kóda. Prilikom definisanja ove kontrole mora se navesti njeno „ime“ (kako bi se znalo kome se prosleđuju podaci iz promenljive ds) i da se sama kontrola izvršava na strani servera. Ona će, na osnovu prosleđenih joj parametara, generisati završni html-fajl; opseg ovih parametara je najviše ograničen funkcionalnošću završne stranice koja će se videti u browser-u. Naime, program nas neće ograničiti u broju kolona u kojima ćemo ispisati rezultat (u



našem primeru je RepeatColumns="2"), ali će izgled stranice biti nepregledan; slično je i sa ostalim parametrima.

Od reda 23 do reda 38 se definiše izgled i pozicija gde treba prikazati podatke locirane u promenljivoj ds. Delovima kóda tipa DataBinder.Eval(Container.DataItem, "naziv_polja") se ispisuje sadržaj kolone konkretnog reda. Ovde se za svaku kolonu navodi naziv same kolone, jer su upitom (linija kóda 12) iz baze „izvučene“ sve kolone, a na samoj stranici prikazujemo samo one kojima smo konkretno napisali ime.

Napomenimo za kraj da se ovde u kódu nigde ne vrši prelazak u naredni zapis rezultata sql-upita, kao što smo imali u ASP-primeru (konkretno RS.MoveNext). Naime, greška koja je nastajala kada bi programer zaboravio da upiše komandu za prelazak u naredni zapis dovođila je do privremene blokade web-servera; da bi se to izbeglo, Microsoft u .NET-u uvodi rešenje kojim se (i) taj proces automatizuje.