

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

آموزش اتصال داده ها / ASP.NET – Data Binding

مدرس : مهندس افشین رفوآ

آموزش اتصال داده ها / ASP.NET – Data Binding

تمامی **control** های فرم وب **ASP.NET** تابع **DataBind** را از کلاس **Control** پدر خود به ارث می برند. متد نام برده به کنترل **web form** یک قابلیت ذاتی اعطا می کند که از طریق آن قادر خواهد بود داده به خاصیت های (**property**) خود متصل کند. از فرایند ذکر شده با نام **simple data binding** (اتصال ساده ی داده) یا **inline data binding** (اتصال درون خطی داده) یاد می شود.

Simple data binding شامل الصاق (متصل) کردن مجموعه آیتم هایی که رابط (**interface**) **IEnumerable** را پیاده سازی می کند یا کلاس های **DataSet** و **DataTable** به خاصیت (**property**) **DataSource** کنترل مورد نظر می باشد.

از سوی دیگر، برخی از کنترل ها قادر هستند رکورد ها، لیست ها یا ستون هایی از داده را به واسطه ی کنترل **DataSource** به درون ساختارهایشان متصل کنند. کنترل های ذکر شده از کلاس **BaseDataBoundControl** مشتق می شوند. از این نوع اتصال داده با نام **declarative data binding** (اتصال اعلانی داده) یاد می شود.

کنترل های **data source** به کنترل های **data bound** کمک می کنند قابلیت ها و امکاناتی همچون مرتب سازی (**sorting**)، صفحه بندی (**paging**) و ویرایش (**editing**) مجموعه داده ها / **data collection** را پیاده سازی کنند.

BaseDataBoundControl یک کلاس انتزاعی (**abstract class**) است که خود توسط دو کلاس انتزاعی دیگر به ارث گرفته می شود، این دو کلاس عبارتند از:

DataBoundControl

HierarchicalDataBoundControl

کلاس انتزاعی **DataBoundControl** خود باز توسط دو کلاس انتزاعی دیگر به ارث گرفته می شود:

ListControl

CompositeDataBoundControl

کنترل هایی که از **simple data binding** پشتیبانی می کنند در حقیقت از کلاس انتزاعی **ListControl** مشتق می شوند. این کلاس ها عبارتند از:

BulletedList

CheckBoxList

DropDownList

ListBox

RadioButtonList

حال کنترل هایی که از **declarative data binding** (که شکل پیچیده و پیشرفته تری از اتصال داده به شمار می رود) پشتیبانی می کنند خود از کلاس انتزاعی **CompositeDataBoundControl** مشتق می شوند. کنترل های مزبور عبارتند از:

DetailsView

FormView

GridView

RecordList

Simple data binding (اتصال ساده ی داده)

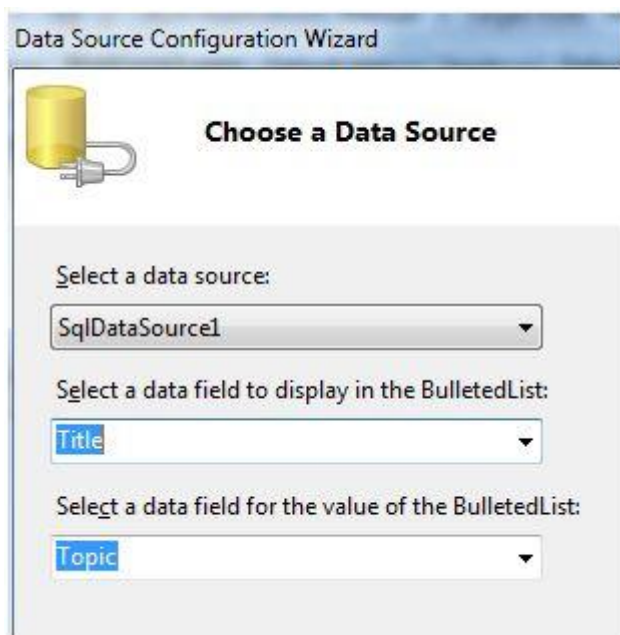
Simple data binding شامل لیست های گزینش فقط خواندنی / **read-only selection list** می باشد. این کنترل ها می توانند به فیلدها یا لیست آرایه از پایگاه داده متصل شوند. لیست های گزینش (**selection list**) دو مقدار از پایگاه یا منبع داده می گیرند؛ یک مقدار توسط خود لیست نمایش داده می شود و دیگری مقداری است که به نمایش (**display**) مربوط می شود.

برای درک هرچه بهتر این مفهوم به ایجاد یک وب سایت که قابلیت های فوق در آن بکار گرفته می شود خواهیم پرداخت.

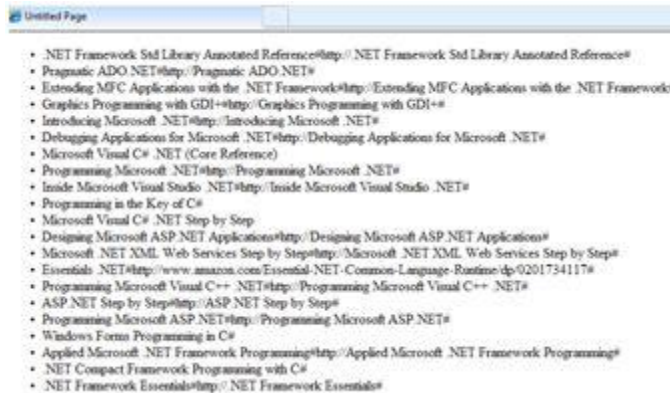
ابتدا یک وب سایت ایجاد کنید که فهرست علامت گذاری شده و مرتب (bulleted list) و همچنین کنترل **SqlDataSource** در آن بکار رفته باشد. کنترل **data source** را طوری پیکربندی کنید که دو مقدار را از پایگاه داده بازیابی کند.

انتخاب یک منبع داده (data source) برای کنترل **bulleted list** شامل موارد زیر می شود:

- انتخاب کنترل **data source**
- انتخاب یک فیلد برای نمایش که **data field** یا فیلد داده خوانده می شود
- انتخاب یک فیلد ویژه ی مقدار مورد نظر



حین اجرای **application** دقت کنید که کل ستون عنوان (**title column**) به لیست علامت گذاری شده (**bulleted list**) متصل شده و نمایش داده می شود.



Declarative Data Binding (اتصال داده به صورت اعلانی)

اگر به مبحث پیشین مراجعه کنید متوجه می شوید که همراه با استفاده از کنترل **GridView**، قابلیت **declarative data binding** را نیز پیاده کرده و بکار بردیم. کنترل های **data bound** مرکب¹ دیگری که قادر هستند داده ها را جدول گونه (**tabular**) نمایش داده و دستکاری کنند عبارتند از: **FormView**، **DetailsView** و **RecordList**.

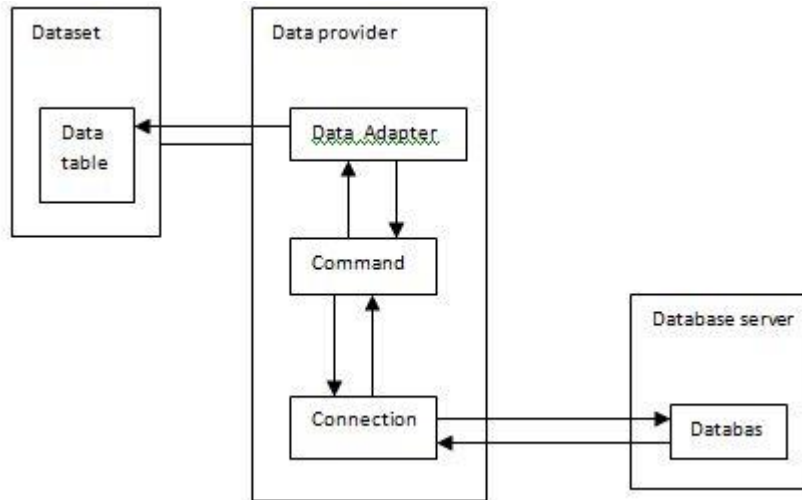
در فصل بعدی به تکنولوژی مدیریت پایگاه داده که **ADO.NET** نامیده می شود نیز خواهیم پرداخت.

فرایند اتصال داده (**data binding**) شامل اشیا زیر می باشد:

- یک **data set** (مجموعه داده) که اطلاعات و داده های بازیابی شده از پایگاه داده را در خود ذخیره می کند.
- یک **data provider** (ارائه دهنده ی داده) که به وسیله ی یک دستور (**command**) از طریق / در طی یک اتصال (**connection**) داده ها را از پایگاه داده ی مورد نظر بازیابی می کند.
- **Data adapter** (به مثابه ی یک پل ارتباطی بین یک منبع داده و یک کلاس داده منفصل یا قطع شده مانند **Dataset** عمل می کند) که دستور **select**، ذخیره شده در شیء **command** را صادر می کند. **Data adaptor** همچنین قادر است با صدور و استفاده از دستوراتی همچون **insert**، **update** و **delete** اطلاعات موجود در پایگاه داده را بروز رسانی کند.

رابطه ی بین اشیا اتصال داده (**data binding objects**):

¹ Composite data bound controls



مثال:

مراحل زیر را دنبال کنید:

مرحله ی اول: یک وب سایت جدید ایجاد کنید. حال یک کلاس به نام **booklist** به آن اضافه کنید. می توانید این کار را

با راست کلیک روی اسم **solution** در پنجره ی **solution explorer** و انتخاب آیتم **'Class'** از پنجره ی محاوره ی **'Add'**

'Item' انجام دهید. کلاس انتخابی را **booklist.cs** نام گذاری کنید:

```

using System;
using System.Data;
using System.Configuration;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
namespace databinding
{
    public class booklist
    {
        protected String bookname;
        protected String authorname;
        public booklist(String bname, String aname)
        {
            this.bookname = bname;
            this.authorname = aname;
        }
        public String Book
        {
            get
            {
                return this.bookname;
            }
            set
            {
                this.bookname = value;
            }
        }
    }
}

```



```

<tr>
<td style="width: 228px; height: 40px;">
<asp:Label ID="lbllistbox" runat="server"></asp:Label>
</td>
<td style="height: 40px">
<asp:Label ID="lbldrpdwn" runat="server">
</asp:Label>
</td>
</tr>
<tr>
<td style="width: 228px; height: 21px"></td>
<td style="height: 21px"></td>
</tr>
<tr>
<td style="width: 228px; height: 21px">
<asp:RadioButtonList ID="RadioButtonList1" runat="server"
AutoPostBack="True" OnSelectedIndexChanged="RadioButtonList1_SelectedIndexChanged">
</asp:RadioButtonList>
</td>
<td style="height: 21px">
<asp:CheckBoxList ID="CheckBoxList1" runat="server"
AutoPostBack="True" OnSelectedIndexChanged="CheckBoxList1_SelectedIndexChanged">
</asp:CheckBoxList>
</td>
</tr>
<tr>
<td style="width: 228px; height: 21px">
<asp:Label ID="lbldrlist" runat="server">
</asp:Label>
</td>
<td style="height: 21px">
<asp:Label ID="lbldchclist" runat="server">
</asp:Label>
</td>
</tr>
</table>
</div>
</form>

```

مرحله ی سوم: در مرحله ی آخر به نوشتن **code behind routine** های اپلیکیشن بپردازید:

```

public partial class WebForm1 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        IList bklist = createbooklist();

        if (!this.IsPostBack)
        {
            this.ListBox1.DataSource = bklist;
            this.ListBox1.DataTextField = "Book";
            this.ListBox1.DataValueField = "Author";

            this.DropDownList1.DataSource = bklist;
            this.DropDownList1.DataTextField = "Book";
            this.DropDownList1.DataValueField = "Author";

            this.RadioButtonList1.DataSource = bklist;
            this.RadioButtonList1.DataTextField = "Book";
            this.RadioButtonList1.DataValueField = "Author";

            this.CheckBoxList1.DataSource = bklist;
            this.CheckBoxList1.DataTextField = "Book";
            this.CheckBoxList1.DataValueField = "Author";
            this.DataBind();
        }
    }
}

```

```

}
protected IList createbooklist()
{
    ArrayList allbooks = new ArrayList();
    booklist bl;
    bl = new booklist("UNIX CONCEPTS", "SUMITABHA DAS");
    allbooks.Add(bl);
    bl = new booklist("PROGRAMMING IN C", "RICHI KERNIGHAN");
    allbooks.Add(bl);
    bl = new booklist("DATA STRUCTURE", "TANENBAUM");
    allbooks.Add(bl);
    bl = new booklist("NETWORKING CONCEPTS", "FOROUZAN");
    allbooks.Add(bl);
    bl = new booklist("PROGRAMMING IN C++", "B. STROUSTROUP");
    allbooks.Add(bl);
    bl = new booklist("ADVANCED JAVA", "SUMITABHA DAS");
    allbooks.Add(bl);
    return allbooks;
}
protected void ListBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    this.lbltextbox.Text = this.ListBox1.SelectedValue;
}

protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    this.lblrdpdown.Text = this.DropDownList1.SelectedValue;
}

protected void RadioButtonList1_SelectedIndexChanged(object sender, EventArgs e)
{
    this.lblrdlist.Text = this.RadioButtonList1.SelectedValue;
}

protected void CheckBoxList1_SelectedIndexChanged(object sender, EventArgs e)
{
    this.lblchklist.Text = this.CheckBoxList1.SelectedValue;
}
}

```

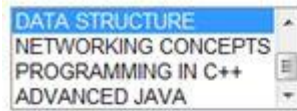
به مسائل زیر توجه کنید:

کلاس **booklist** دارای دو خاصیت می باشد: **bookname** و **authname**

تابع **createbooklist** یک متد **user-defined** (تعریف شده توسط کاربر) می باشد که آرایه ای از اشیا **booklist** به نام **allbooks** ایجاد می کند.

مدیریت کننده ی رخداد **Page_Load (event handler)** فهرستی از کتاب ها ایجاد می کند. لیست مورد نظر از نوع **IList** است . این لیست رابط **IEnumerable (interface)** را پیاده سازی کرده و توانایی متصل بودن به کنترل های **list** را دارا می باشد. مدیریت کننده ی رخداد **Page_Load** شیء **IList** 'bklist' را به کنترل های **list** متصل می کند. خاصیت **bookname (property)** نمایش داده شده و خاصیت **authname** مقدار در نظر گرفته می شود.

هنگامی که صفحه اجرا می شود، اگر کاربر یک کتاب انتخاب کند، اسم کتاب مورد نظر انتخاب شده و در کنترل های **list** نمایش داده می شود در حالی که **label** های مربوطه اسم نویسنده ی کتاب را نمایش می دهند.



DATA STRUCTURE

TANENBAUM

- UNIX CONCEPTS
- PROGRAMMING IN C
- DATA STRUCTURE
- NETWORKING CONCEPTS
- PROGRAMMING IN C++
- ADVANCED JAVA

FOROUZAN

TANENBAUM

- UNIX CONCEPTS
- PROGRAMMING IN C
- DATA STRUCTURE
- NETWORKING CONCEPTS
- PROGRAMMING IN C++
- ADVANCED JAVA

SUMITABHA DAS