

Projektowanie aplikacji ASP.NET
Wykłady 06-07/15
ASP.NET WebForms DataBinding

Wiktor Zychła 2019/2020

Spis treści

1	Rodzaje wiązania danych (ang. data binding)	2
2	Przegląd mechanizmów szablonowania.....	3
3	Przykład	4

1 Rodzaje wiązania danych (ang. data binding)

Istnieją dwa możliwe podejścia do prezentacji zbioru danych (bez względu na rodzaj komponentu – ta sama zasada obowiązuje dla list rozwijalnych czy dla gridów):

- Wiązanie imperatywne – w kodzie strony w części code behind (kod C#)
- Wiązanie deklaratywne – w kodzie strony w części deklaratywnej (*.aspx)

Oba rodzaje mają swoje plusy i minusy, generalnie jednak WebForms kładzie duży nacisk na wiązanie deklaratywne.

2 Przegląd mechanizmów szablonowania

WebForms obsługuje szereg tzw. *bee stings* umożliwiających proste szablonowanie zawartości. Szablonowanie jest uproszczonym wiązaniem, ponieważ nie obsługuje wiązania dwukierunkowego.

- `<% %>` - is for [inline code](#) (especially logic flow)
- `<%$ %>` - is for [evaluating expressions](#) (like resource variables)
- `<%@ %>` - is for [Page directives](#), registering assemblies, importing namespaces, etc.
- `<%= %>` - is short-hand for `Response.Write` (discussed [here](#))
- `<%# %>` - is used for [data binding expressions](#).
- `<%: %>` - is short-hand for `Response.Write(Server.HtmlEncode())` ASP.net 4.0+
- `<%#: %>` - is used for [data binding expressions](#) and is automatically HTMLEncoded.
- `<%-- --%>` - is for [server-side comments](#)

(za <https://stackoverflow.com/a/957321/941240>)

Proste szablonowanie za pomocą `<%= %>` lub `<%: %>` działa wyłącznie wtedy kiedy zawartość jest wstrzykiwana do takiej zawartości strony która nie jest kontrolką renderowaną po stronie serwera.

Oznacza to że zadziała

```
<input type="text" value="<%= Example() %>" />
```

ale nie zadziała

```
<asp:TextBox runat="server" value="<%= Example() %>" />
```

W tym drugim przypadku konieczne jest użycie wyrażenia wiązania postaci `<%# ... %>`.

3 Przykład

W przykładzie rozważymy model danych, odwołujący się wyłącznie do pamięci (aby nie zaciemniać przykładu):

```
public class Miejscowosc
{
    public Miejscowosc()
    {
    }

    public int iD;
    public int ID
    {
        get { return iD; }
        set { iD = value; }
    }

    public string nazwa;
    public string Nazwa
    {
        get { return nazwa; }
        set { nazwa = value; }
    }
}
```

```
public class Osoba : IFormattable
{
    public Osoba()
    {
    }

    public int iD;
    public int ID
    {
        get { return iD; }
        set { iD = value; }
    }

    public int iD_MIEJSCOWOSC;
    public int ID_MIEJSCOWOSC
    {
        get { return iD_MIEJSCOWOSC; }
        set { iD_MIEJSCOWOSC = value; }
    }

    public string nazwisko;
    public string Nazwisko
    {
        get { return nazwisko; }
        set { nazwisko = value; }
    }

    public string imie;
    public string Imie
    {

```

```

        get { return imie; }
        set { imie = value; }
    }

    public Miejscowosc MiejscowoscUr
    {
        get
        {
            foreach (Miejscowosc m in Model.Instance.Miejscowosci)
                if (m.ID == this.ID_MIEJSCOWOSC)
                    return m;

            return null;
        }
        set
        {
            ID_MIEJSCOWOSC = value.ID;
        }
    }

    public Osoba This
    {
        get
        {
            return this;
        }
    }

    #region IFormattable Members

    public string ToString(string format, IFormatProvider formatProvider)
    {
        switch (format)
        {
            case "u" :
                return string.Format("{0} {1} [ur. {2}]", Nazwisko, Imie, MiejscowoscUr.Nazwa );
            default :
                return string.Format("{0} {1}", Nazwisko, Imie);
        }
    }

    #endregion
}

```

```

public class Model
{
    public List<Osoba> Osoby = new List<Osoba>();
    public List<Miejscowosc> Miejscowosci = new List<Miejscowosc>();

    public Model()
    {
        for (int i = 0; i < 100; i++)
        {
            Osoba o = new Osoba();

            o.ID = i;
            o.ID_MIEJSCOWOSC = i % 25;
        }
    }
}

```

```

        o.Imie = "Imie " + i.ToString();
        o.Nazwisko = "Nazwisko " + i.ToString();

        Osoby.Add(o);
    }

    for (int j = 0; j < 25; j++)
    {
        Miejscowosc m = new Miejscowosc();

        m.ID = j;

        m.Nazwa = "Miejscowosc " + j.ToString();

        Miejscowosci.Add(m);
    }
}

public static Model Instance
{
    get
    {
        if (HttpContext.Current.Application["model"] == null)
        {
            Model model = new Model();
            HttpContext.Current.Application["model"] = model;
        }

        return (Model)HttpContext.Current.Application["model"];
    }
}
}
}

```

```

public class Osoba_DataProvider
{
    public Osoba_DataProvider()
    {
    }

    /// <summary>
    /// Metoda dla źródła danych GridView
    ///
    /// Parametry OrderBy, StartRow i RowCount zdefiniowane są
    /// w odpowiednich właściwościach źródła danych
    /// </summary>
    /// <param name="OrderBy"></param>
    /// <param name="StartRow"></param>
    /// <param name="RowCount"></param>
    /// <returns></returns>
    public List<Osoba> Retrieve( string OrderBy, int StartRow, int RowCount)
    {
        if (RowCount > 0)
            return Model.Instance.Osoby.GetRange(StartRow, RowCount);
        else
            return new List<Osoba>(new Osoba[] { Model.Instance.Osoby[StartRow]
    } );
    }
}

```

```

/// <summary>
/// jw
/// </summary>
/// <returns></returns>
public int SelectCount()
{
    return Model.Instance.Osoby.Count;
}

/// <summary>
/// Metoda dla źródła danych DetailsView
///
/// Parametr ID zdefiniowany jest w sekcji SelectParameters źródła danych
/// </summary>
/// <param name="ID"></param>
/// <returns></returns>
public Osoba Retrieve( int ID )
{
    return Model.Instance.Osoby
        .Find( delegate ( Osoba osoba ) { return osoba.ID == ID; } );
}
}

```

Dla tak zdefiniowanego modelu możliwa jest ilustracja mechanizmów wiązania deklaratywnego, nie wymagających w ogóle elementów imperatywnych poza jednym wyjątkiem – wiązania listy rozwijalnej która dla ilustracji nie jest powiązana ze źródłem danych implementujących interfejs **IDataList**:

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inheri
ts="_Default" %>

<!DOCTYPE html>

<html>
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            Databinding formantu DropDownList:<br />
            <asp:DropDownList ID="DropDownList1" runat="server"
                DataSource='<%# Model.Instance.Osoby %>' DataValueField="ID"
                DataTextField="This" DataTextFormatString="{0:u}">
            </asp:DropDownList>
            <asp:Button ID="Button1" runat="server" Text="Button" /><br />
            <br />
            Databinding formantu GridView ze stronicowaniem i ObjectDataSource:<br />
            <asp:GridView ID="GridView1" runat="server" AllowPaging="True" AllowSort
ing="True"
                AutoGenerateColumns="False" CellPadding="4" DataSourceID="ObjectData
Source1"
                ForeColor="#333333" GridLines="None">
                <FooterStyle BackColor="#5D7B9D" Font-
Bold="True" ForeColor="White" />
                <RowStyle BackColor="#F7F6F3" ForeColor="#333333" />

```

```

        <EditRowStyle BackColor="#999999" />
        <SelectedRowStyle BackColor="#E2DED6" Font-
Bold="True" ForeColor="#333333" />
        <PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="C
enter" />
        <HeaderStyle BackColor="#5D7B9D" Font-
Bold="True" ForeColor="White" />
        <AlternatingRowStyle BackColor="White" ForeColor="#284775" />
        <Columns>
            <asp:TemplateField HeaderText="Nazwisko">
                <ItemTemplate>
                    <%# ((Osoba)Container.DataItem).Nazwisko %>
                </ItemTemplate>
            </asp:TemplateField>
            <asp:TemplateField HeaderText="Imie">
                <ItemTemplate>
                    <%# ((Osoba)Container.DataItem).Imie %>
                </ItemTemplate>
            </asp:TemplateField>
            <asp:TemplateField HeaderText="Miejscowość ur." SortExpression="M
SCUR">
                <ItemTemplate>
                    <%# ((Osoba)Container.DataItem).MiejscowoscUr.Nazwa %>
                </ItemTemplate>
            </asp:TemplateField>
        </Columns>
    </asp:GridView>
    <asp:ObjectDataSource ID="ObjectDataSource1" runat="server" EnablePaging
="True" InsertMethod="Insert"
        MaximumRowsParameterName="RowCount" SelectCountMethod="SelectCount"
SelectMethod="Retrieve"
        SortParameterName="OrderBy" StartRowIndexParameterName="StartRow" Ty
peName="Osoba_DataProvider"
        UpdateMethod="Update"></asp:ObjectDataSource>
    <br />
    Databinding formantu DetailsView ze stronicowaniem i ObjectDataSource<br
/>
    <asp:DetailsView ID="DetailsView1" runat="server" Height="50px" Width="1
00%" AutoGenerateDeleteButton="True"
        AutoGenerateEditButton="True" AutoGenerateInsertButton="True" AutoGe
nerateRows="False" CellPadding="4"
        DataSourceID="ObjectDataSource2" ForeColor="#333333" GridLines="None
" AllowPaging="True">
        <FooterStyle BackColor="#5D7B9D" Font-
Bold="True" ForeColor="White" />
        <CommandRowStyle BackColor="#E2DED6" Font-Bold="True" />
        <EditRowStyle BackColor="#999999" />
        <RowStyle BackColor="#F7F6F3" ForeColor="#333333" />
        <PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="C
enter" />
        <Fields>
            <asp:TemplateField HeaderText="Nazwisko">
                <ItemTemplate>
                    <%# ((Osoba)Container.DataItem).Nazwisko %>
                </ItemTemplate>
            </asp:TemplateField>
            <asp:TemplateField HeaderText="Imie">
                <ItemTemplate>
                    <%# ((Osoba)Container.DataItem).Imie %>
                </ItemTemplate>
            </asp:TemplateField>
        </Fields>
    </asp:DetailsView>

```



```

        </ItemTemplate>
    </asp:TemplateField>
    <asp:TemplateField HeaderText="Miejscowość ur.">
        <ItemTemplate>
            <%# ((Osoba)Container.DataItem).MiejscowoscUr.Nazwa %>
        </ItemTemplate>
    </asp:TemplateField>
</Fields>
<FieldHeaderStyle BackColor="#E9ECF1" Font-
Bold="True" Width="20%" />
<HeaderStyle BackColor="#5D7B9D" Font-
Bold="True" ForeColor="White" />
<AlternatingRowStyle BackColor="White" ForeColor="#284775" />
</asp:DetailsView>
<asp:ObjectDataSource ID="ObjectDataSource2" runat="server" SelectMethod
="Retrieve" TypeName="Osoba_DataProvider"
    EnablePaging="True" MaximumRowsParameterName="StartRow" StartRowInde
xParameterName="RowCount"
    SelectCountMethod="SelectCount" SortParameterName="OrderBy"></asp:Ob
jectDataSource>
<br />
Databinding formantu DetailsView bez stronicowania ale ze wskazaniem kon
kretnego
rekordu z ObjectDatasource<br />
<br />ID rekordu: <asp:TextBox ID="TextBox1" runat="server">5</asp:TextB
ox><br />
<br />
<asp:DetailsView ID="DetailsView2" runat="server" Height="50px" Width="1
00%" AutoGenerateDeleteButton="True"
    AutoGenerateEditButton="True" AutoGenerateInsertButton="True" AutoGe
nerateRows="False" CellPadding="4"
    DataSourceID="ObjectDataSource3" ForeColor="#333333" GridLines="None
" AllowPaging="True">
    <FooterStyle BackColor="#5D7B9D" Font-
Bold="True" ForeColor="White" />
    <CommandRowStyle BackColor="#E2DED6" Font-Bold="True" />
    <EditRowStyle BackColor="#999999" />
    <RowStyle BackColor="#F7F6F3" ForeColor="#333333" />
    <PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="C
enter" />
</Fields>
    <asp:TemplateField HeaderText="Nazwisko">
        <ItemTemplate>
            <%# ((Osoba)Container.DataItem).Nazwisko %>
        </ItemTemplate>
    </asp:TemplateField>
    <asp:TemplateField HeaderText="Imie">
        <ItemTemplate>
            <%# ((Osoba)Container.DataItem).Imie %>
        </ItemTemplate>
    </asp:TemplateField>
    <asp:TemplateField HeaderText="Miejscowość ur.">
        <ItemTemplate>
            <%# ((Osoba)Container.DataItem).MiejscowoscUr.Nazwa %>
        </ItemTemplate>
    </asp:TemplateField>
</Fields>
<FieldHeaderStyle BackColor="#E9ECF1" Font-
Bold="True" Width="20%" />

```

```

        <HeaderStyle BackColor="#5D7B9D" Font-
Bold="True" ForeColor="White" />
        <AlternatingRowStyle BackColor="White" ForeColor="#284775" />
    </asp:DetailsView>
    <asp:ObjectDataSource ID="ObjectDataSource3" runat="server" SelectMethod
="Retrieve"
        TypeName="Osoba_DataProvider" MaximumRowsParameterName="" StartRowIn
dexParameterName="">
        <SelectParameters>
            <asp:ControlParameter ControlID="TextBox1" Name="ID" />
        </SelectParameters>
    </asp:ObjectDataSource>
</div>
</form>
</body>
</html>

```

The screenshot shows a web browser window with the URL `http://localhost:49786/Default.aspx`. The browser displays three distinct data binding scenarios:

Databinding formantu DropDownList:
 A dropdown menu is shown with the text "Nazwisko 0 Imie 0 [ur. Miejscowosc 0]" and a "Button" next to it.

Databinding formantu GridView ze stronicowaniem i ObjectDataSource:
 A GridView is displayed with columns "Nazwisko", "Imie", and "Miejscowość ur.". The rows are numbered from 0 to 9. A blue bar at the bottom of the grid contains the numbers "12345678910".

Databinding formantu DetailsView ze stronicowaniem i ObjectDataSource:
 A DetailsView is shown with fields for "Nazwisko", "Imie", and "Miejscowość ur.". The values displayed are "Nazwisko 1", "Imie 1", and "Miejscowosc 1". Below the fields are links for "Edytuj", "Usuń", and "Nowy". A blue bar at the bottom contains the numbers "12345678910...".

Databinding formantu DetailsView bez stronicowania ale ze wskazaniem konkretnego rekordu z ObjectDataSource:
 A DetailsView is shown with a text input field for "ID rekordu:" containing the value "5". Below it are fields for "Nazwisko", "Imie", and "Miejscowość ur." with values "Nazwisko 5", "Imie 5", and "Miejscowosc 5". Links for "Edytuj", "Usuń", and "Nowy" are also present.