Projektowanie aplikacji ADO.NET + ASP.NET
Zestaw 3

Data Binding

23-10-2012

Liczba punktów do zdobycia: **10/30**
Zestaw ważny do: 20-11-2012

1. (**1p**) Zwiąż dowolny formant wiązalny, np. `DropDownList` z relacyjnym źródłem danych (dowolny `IListSource` np. `DataSet, DataTable, DataView`) a potem z obiektowym (dowolny `IEnumerable`, np. `ArrayList, List<T>`).

   Jak określać formatowanie wartości dla typów prostych na poziomie deklaratywnym (przykład: na liście rozwijalnej mają pokazać się wartości typu `DateTime` z formatowaniem `"d"`?

   Jak określać formatowanie wartości dla typów referencyjnych na poziomie deklaratywnym (przykład: obiekt implementuje interfejs `IFormattable` i chcę żeby na liście rozwijalnej pokazał się z formatowaniem `"d"`?

2. (**1p**) Pokaż w jaki sposób wiązać dane do formantu `GridView`.

   Na czym polega problem z mechanizmem stronicowania oferowanym przez zwykły `SqlDataSource` i dlaczego jest on niewydajny?

3. (**1p**) Użyj `ObjectDataSource` jako pośrednika do źródła danych dla `GridView` i pokaż że dzięki temu możliwe jest zaimplementowanie wydajnych mechanizmów stronicowania. Pokaż jak `ObjectDataSource` otrzymuje i wykorzystuje informację o aktualnym porządku sortowania.

4. (**1p**) Naucz się budować szablony kolumn (`asp:TemplateField`) w trybie podglądu i edycji dla `GridView`.

   Załóżmy taką relację między dwoma klasami biznesowymi Child i Parent, w której każdy element Child wskazuje jednoznacznie na element Parent.

   Zilustruj przykładem kodu scenariusz wiązania się do siatki obiektów Child w taki sposob, aby:

   - w trybie podglądu w kolumnie Master można było zobaczyć literalną reprezentację obiektu Parent (Parent.Nazwa)
   - w trybie edycji w kolumnie Parent użytkownik dostawał do dyspozycji listę rozwijalną elementów z kategorii Parent, a w trybie edycji lista ta dodatkowo pozycjonowała by się na tym obiekcie Parent na który wskazuje aktualnie edytowany Child

5. (**1p**) Naucz się korzystać z jakiegokolwiek innego dostawcy danych, np. `XmlDataSource`, `LinqDataSource`

6. (**2p**) Zauważ, że listę parametrów metody `SelectMethod` z `ObjectDataSource` można rozszerzyć o dowolne parametry zadeklarowane w sekcji `SelectParameters` w deklaracji źródła danych.

   Przykładowo, w ten sposób wskazuje się na przykład parametr oznaczający identyfikator rekordu, który zamierzamy pokazać w `DetailsView`.

   Lista dostępnych źródeł wartości tych dodatkowych parametrów obejmuje m.in. wartości z ciastek, sesji, z listy parametrów, czy z formantów. ASP.NET daje jednak możliwość dostarczenia własnej logiki podawania wartości parametrów. Wzorując się na przykładzie *Creating Custom Parameter Objects*:

   `http://www.java2s.com/Code/ASP/ADO.net-Database/CreatingCustomParameterObjects.htm`

   zaimplementuj klasę umożliwiającą przesyłanie parametrów w postaci zaszyfrowanej w parametrach `QueryString` żądania.

   Formalnie: załóżmy że przez `QueryString` przekazujemy parametry typu `string`. Klasa parametru powinna dostarczać pomocniczej metody, za pomocą której można zaszyfrować wartość, tak żeby można umieścić ją w `QueryString`.

```
string MyValue     = "Parametr do przekazania";
string QueryString = string.Format( "customparam={0}", EncryptedParameter.Encrypt( MyValue ) );
```

   Następnie należy zadbać o to, by metoda `Evaluate` klasy parametru potrafiła wydobyć z `QueryString` odpowiednią wartość, zdeszyfrować ją i przekazać do klasy dostawcy danych (*data provider*) w postaci jawnej.

```
<asp:ObjectDataSource
    id="TheDataSource"
    TypeName="TheNamespace.CustomDataProvider"
    SelectMethod="Retrieve"
    Runat="server">
    <InsertParameters>
        <custom:EncryptedParameter Name="customparam" QueryStringField="customparam" />
    </InsertParameters>
</asp:ObjectDataSource>
```

7. (**1p**) Naucz się korzystać z kolumny typu `asp:CommandField` formantu `GridView`.

8. (**1p**) Pokaż w jaki sposób przechwytywać zdarzenia przycisków zagnieżdżonych wewnątrz `GridView` w zdarzeniu `RowCommand`.

   - Poczytaj o tym jakie rodzaje akcji są standardowo przechwytywane na poziomie `RowCommand`.

     `http://msdn2.microsoft.com/en-us/library/`
     `system.web.ui.webcontrols.gridview.rowcommand.aspx`

   - Zdefiniuj w jednej z kolumn własny przycisk typu `asp:ImageButton` i zadaj mu parametry tak, aby po stronie `RowCommand` otrzymywać informację o kliknięciu Twojego przycisku z argumentem określającym identyfikator obiektu znajdującego się w kliknięty wierszu.

9. (**1p**) Pokaż w jaki sposób dodać potwierdzenie usunięcia wiersza dla `GridView`, tzn. po kliknięciu przycisku "Usuń" użytkownik powinien zobaczyć w przeglądarce okno ("Czy na pewno usunąć element?") z przyciskami "OK" i "Anuluj".

Wiktor Zychla