

Projektowanie aplikacji ADO.NET + ASP.NET

Zestaw 3

Data Binding

27-10-2009

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

Jak zadawać formatowanie wartości dla typów prostych na poziomie deklaratywnym? Jak zadawać formatowanie wartości dla typów referencyjnych na poziomie deklaratywnym?

... czyli co i gdzie wpisać w atrybuty formantu po stronie deklaratywnej, żeby uzyskać na przykład formatowanie dat formatem daty krótkiej albo formatowanie dowolnego obiektu formatem określonym w implementacji przez ten obiekt interfejsu `IFormattable`

2. (2p) 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?

Użyj `ObjectDataSource` jako pośrednika do źródła danych i pokaż że dzięki temu możliwe jest zaimplementowanie wydajnych mechanizmów stronicowania. Pokaż jak `ObjectDataSource` otrzymuje i wykorzystuje informację o aktualnym porządku sortowania.

3. (2p) Naucz się budować szablony kolumn (`asp:TemplateField`) w trybie podglądu, edycji i dodawania dla `GridView`.

Zalóżmy taką relację między dwoma klasami biznesowymi `Child` i `Master`, w której każdy element `Child` wskazuje jednoznacznie na element `Master` (na przykład osoby i miejscowości urodzenia).

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

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

4. (1p) Naucz się korzystać z jakiegokolwiek innego dostawcy danych, np. `XmlDataSource`, `LinqDataSource`

5. (2p) Zauważ, że listę parametrów metody `SelectMethod` z `ObjectDataSource` można rozszerzyć o dowolne parametry zadeklarowane w sekcji `SelectParameters` w deklaracji

źródła danych. 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:

```
http://www.leftslipper.com/ShowFaq.aspx?FaqId=11
```

zaimplementować klasę dostarczającą wartość parametru będącą konkatencją wszystkich parametrów z `QueryString` bieżącego żądania.

Uwaga! Możliwość określania własnej logiki dla parametrów źródła danych jest bardzo przydatna. Na przykład, wartość parametru ID dla `DetailsView` może być przekazana w liście parametrów `QueryString` w postaci zaszyfowanej, niemożliwej do zniekształcenia w przeglądarce, a odszyfrowaniem zajmuje się wtedy klasa, będąca własnym parametrem.

6. (1p) Naucz się korzystać z kolumny typu `asp:CommandField` formantu `GridView`.
7. (1p) Pokaż w jaki sposób dodać potwierdzenie usunięcia wiersza dla `GridView` i to zarówno w przypadku przycisku usuwania z `asp:CommandField` jak i jakiegokolwiek przycisku utworzonego przez `asp:ButtonField`.
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 klikniętym wierszu.