

Projektowanie aplikacji ASP.NET

Zestaw 3

Data Binding

29-10-2013

Liczba punktów do zdobycia: **10/30**

Zestaw ważny do: 19-11-2013

1. (**1p**) Zwiąż dowolny formant wiązalny, np. `DropDownList` z relacyjnym źródłem danych (dowolny `IListSource` np. `DataSet`, `DataTable`, `DataView`) a potem z obiektywnym (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"?)

Wskazówka: `DataTextField`, `DataTextFormatString` oraz pokazany na wykładzie sposób z propecją `This` w klasie.

2. (**1p**) Naucz się korzystać z jakiegokolwiek innego niż `ObjectDataSource` dostawcy danych, np. `SqlDataSource`, `XmlDataSource`, `LinqDataSource`.

Uwaga. `ObjectDataSource` będzie wykorzystywany w kolejnych zadaniach i jak powiedziano na wykładzie - z punktu widzenia właściwej architektury, to jest jedyny właściwy sposób organizacji dostępu do danych.

3. (**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](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.
4. (**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".
 5. (**1p**) Pokaż w praktyce co najmniej dwa różne układy formantu stronicującego (`<asp:DataPager>`) dla `ListView`.

6. (2p) Przykład z wykładu, w którym pojawia się formant `ListView` formatowany w postaci tabeli uzupełnij o następujące sposoby formatowania:

- `ListView` w postaci widoku pojedynczego elementu z atrybutami w układzie wierszowym jak `DetailsView` (uwaga, `ObjectDataSource` może mieć jeden dodatkowy parametr tak, żeby odpytywał źródło danych o tylko jeden element) (1p)
- `ListView` w postaci listy wypunktowanej z nagłówkiem w postaci tekstu i kolejno numerowanymi elementami, renredowanymi jako ` ` (1p)

7. (3p) (Sklep internetowy)

Formant `ListView` za pomocą `ObjectDataSource` zwiąż z taką klasą dostawcy danych (data provider), w której implementacje metod obsługujących dane będą używać dowolnej technologii mapowania obiektowo-relacyjnego.

Prezentowany widok ma dotyczyć towarów w sklepie internetowym - każdy towar ma mieć nazwę, opis, cenę i link do obrazka. Zadbaj więc o to, żeby `ListView` miał właściwie opisany szablon elementu (`ItemTemplate`).

Zwróć uwagę na poprawne przekazywanie parametrów **sortowania i wybierania strony** danych do warstwy mapowania obiektowo-relacyjnego w taki sposób, żeby zapytanie wysyłane ostatecznie do serwera bazy danych było optymalne (tzn. wybierało faktycznie **stronę** danych w zadanym **porządku**).

Formant powinien za pomocą mapowania obiektowo-relacyjnego obsługiwać wszystkie **cztery** rodzaje operacji - prezentację, modyfikację, dodawanie i usuwanie. Innymi słowy - widok ma charakter administracyjny, administrator sklepu internetowego nie tylko przeląda listę towarów ale też może każdy z nich wyedytować oraz dodawać nowe.

Uwaga. `ObjectDataSource` powinien odczytywać dane z jakiegoś rzeczywistego systemu bazodanowego. Najszybciej będzie oczywiście użyć technologii oferującej zapytania LINQ.

Wiktor Zychła