

Projektowanie aplikacji ADO.NET + ASP.NET

Zestaw 1

Podstawy ASP.NET

27-09-2011

Liczba punktów do zdobycia: **10/10**
Zestaw ważny do: 17-10-2011

- (1p)** Na serwerze IIS 6.0/7.0/7.5 założyć dwie różne witryny w różnych fizycznych lokalizacjach na dysku. Dla każdej z witryny utworzyć osobną pulę aplikacji. W kodzie serwerowym (C#) spróbować otworzyć do odczytu wskazany plik na dysku (**StreamReader sr = ...**) i przekonać się, że próba jego otwarcia wymaga nadania jawnie w systemie operacyjnym uprawnień do pliku dla konta określającego tożsamość puli aplikacji.

Co się dzieje w przypadku niewystarczających uprawnień? Dlaczego nie jest dobrą praktyką przełączanie tożsamości puli aplikacji na LocalSystem?

Którąś z utworzonych witryn udostępnić na dwóch różnych nagłówkach hosta (np. `ap1.myserver.com` i `ap2.myserver.com`). Pokazać, że witryna działa poprawnie adresowana na dwa różne sposoby. Nauczyć się korzystać z lokalnej mapy hostów (Windows/System32/Drivers/etc/hosts).
- (1p)** W konfiguracji witryny na serwerze aplikacji umieć wskazać miejsce definiowania filtrów kojarzących rozszerzenia zasobów z logiką ich przetwarzania. Co się stanie jeśli na serwerze aplikacji umieścimy zasób o rozszerzeniu, na które nie mapuje się żaden filtr, a następnie spróbujemy do serwera wysłać żądanie o wydanie tego zasobu? Zademonstrować to na przykładzie eksperymentu z zasobem o rozszerzeniu np. `*.foo`.
- (1p)** Nauczyć się korzystać z Microsoft Network Monitor (<http://support.microsoft.com/kb/93374>) do podglądania ruchu klient-serwer na poziomie protokołów transportowych. Pokazać jak podglądać ramki TCP/IP i jak podglądać komunikację za pomocą protokołu HTTP.
- (1p)** Nauczyć się korzystać z debuggera warstwy HTTP, Fiddlera (<http://www.fiddler2.com/>). Pokazać jak za Fiddlera symulować żądania typu GET i żądania typu POST do własnej strony ASP.NET (czyli jak wyklikać w Fiddlerze żądanie, które do naszej aplikacji wyśle Fiddler, a nie przeglądarka!).

Pokazać jak na serwerze wylapuje się parametry przesłane przez GET a jak te przesłane przez POST.
- (2p)** Nauczyć się korzystać z podstawowych formantów ASP.NET (za http://www.w3schools.com/aspnet/aspnet_refwebcontrols.asp). Jakie odpowiedniki HTML mają poszczególne formanty serwerowe? Przygotować tabelkę na zasadzie *Format serwerowy vs Format w przeglądarce*, np. `<asp:TextBox>` - `<input type="text">`.
- (1p)** Po stronie serwera używamy formantów serwerowych (`<asp:...>`), po stronie klienta renderują się one jako formanty HTML. Czy to wyczerpuje paletę możliwości?

Pokazać, że formanty HTML mogą być przetwarzane po stronie serwera (`runat="server"`). Jaki mógłby być z tego pożytek?

Pokazać, że formantom ASP.NET można w kodzie serwerowym dodawać `explicit` zdarzenia po stronie klienta (czyli mieć dużą kontrolę nad tym jak po stronie klienta renderuje się formant serwerowy). Do czego tym razem można wykorzystać tę możliwość?

7. **(3p)** Przygotować aplikację ASP.NET, która pozwala wprowadzić i wydrukować standardowy "pasek zgłoszenia zadań".

Aplikacja ma składać się z dwóch stron: formularza zgłoszenia i formularza wydruku.

Na formularzu zgłoszenia użytkownik aplikacji powinien mieć możliwość wpisania imienia i nazwiska, daty, nazwy zajęć i numeru zestawu oraz kompletu wyników kolejnych deklarowanych 10 zadań z odpowiednią liczbą punktów. Program powinien kontrolować poprawność wpisywanych danych.

Po zaakceptowaniu formularza zgłoszenia, użytkownik powinien w przeglądarce zobaczyć formularz wydruku: pasek zgłoszenia w postaci możliwej do natychmiastowego wydrukowania.

Użyć dowolnej, wybranej przez siebie metody przekazywania danych między stronami (GET, ciastka, kontenery serwerowe).

Wiktor Zychła