

Projektowanie aplikacji ADO.NET + ASP.NET

Zestaw 6

Tworzenie własnych formantów

24-11-2007

1. **(1p)** Utworzyć własną kontrolkę użytkownika (`*.ascx`), `Login`, która będzie zawierać pola tekstowe **Nazwa użytkownika** i **Hasło** oraz przycisk **Loguj**.
Po naciśnięciu przycisku kontrolka powinna symulować uwierzytelnianie użytkownika używając skonfigurowanego dostawy usługi uwierzytelniania i w wypadku pomyślnego uwierzytelnienia przekierowywać kontekst do innej strony lub informować aplikację podnosząc jakieś zdarzenie.
2. **(1p)** Utworzyć własną kontrolkę dziedziczącą z `WebControl`, która w zależności od stanu wartości `HttpContext.Current.User` będzie wypisywać informację **"Niezalogowany"** lub **Witaj: xyz**, gdzie `xyz` będzie nazwą użytkownika odczytaną z `HttpContext.Current.User`
3. **(3p)** Wzorując się na przykładzie z wykładu napisać własny walidator działający po stronie klienta `PeselValidator`, który będzie walidował poprawność numeru PESEL wprowadzonego do pola tekstowego.

Uwaga! Algorytm weryfikacji poprawności numeru PESEL jest opisany w sieci. Proszę również pamiętać o poprawnej implementacji walidacji po stronie serwera, tak aby właściwość `IsValid` zwracała poprawną wartość.

Uwaga! Walidator powinien działać tak, jak każdy inny walidator, tzn. nie dopuszczać do przesłania na serwer formularza w sytuacji niewłaściwej walidacji pola, do którego jest doczepiony. Walidator powinien poprawnie obsługiwać typowe właściwości, takie jak `ErrorMessage` czy `ValidationGroup`.

Wskazówka! Podobnie jak w przykładzie z wykładu głównym elementem walidatora będzie odpowiednio skonstruowana funkcja JavaScript przywiązana do zdarzeń pola tekstowego po stronie klienta.