

Projektowanie aplikacji ADO.NET + ASP.NET

Zestaw 2

Podstawy ASP.NET

20-10-2009

1. (1p) Nauczyć się dodawać, odczytywać i usuwać ciastka (`HttpCookie`) w kodzie po stronie serwera. Jak sprawdzić czy przeglądarka obsługuje ciastka?
2. (1p) Postudiować interfejsy obiektów `Request`, `Server` i `Response`. Do czego służy obiekt `HttpContext.Current`? Jak z `HttpContext.Current` odczytać aktualnie przetwarzaną w potoku stronę?
3. (2p+1) Wykorzystać obiekt sesji do następującego **naiwnego** rozwiązania problemu autentykacji: w każdej stronie aplikacji w wypadku stwierdzenia w kodzie zdarzenia `Page_Load` braku informacji o użytkowniku w sesji, kontekst przetwarzania przekierowywany jest do strony `Login.aspx`, na której po poprawnym potwierdzeniu tożsamości (login i hasło) informacja o tożsamości zapamiętywana jest w sesji.

Dodatkowy punkt za rozwiązanie, w którym punkt wejścia do aplikacji zostanie poprawnie przywrócony po autentykacji (użytkownik wpisuje adres którejkolwiek strony, jest przekierowany do strony logowania, a następnie aplikacja sama powraca do strony, od której użytkownik chciał rozpocząć nawigację).

*Uwaga! O tym jak **poprawnie** implementować mechanizm autentykacji i autoryzacji będziemy rozmawiać na kolejnych wykładach. To zadanie ma wyłącznie pokazać, że referencyjne rozwiązanie, które poznamy w przyszłości, nie jest jedynym możliwym.*

4. (2p) Przećwiczyć w praktyce przesyłanie danych binarnych w obie strony. Ściślej - wykonać aplikację, która pozwoli użytkownikowi wskazać plik lokalny na dysku i przesłać go na serwer (formant `<input type="file" ... />`), a po stronie serwera zostanie wyprodukowany i odesłany plik XML:

```
<opis>
  <nazwa>nazwaprzesłanegopliku</nazwa>
  <rozmiar>rozmiarprzesłanegopliku</rozmiar>
  <sygnatura>sumabajtówplikulo modulo 0xFFFF</sygnatura>
</opis>
```

Plik XML powinien być budowany dynamicznie i odesłany do klienta **bez** zapisywania jego zawartości na dysku serwera. W przeglądarce użytkownika nadesłana odpowiedź powinna spowodować podniesienie się domyślnego okna Otwórz/Zapisz/Anuluj.

5. (1p) Nauczyć się różnic między kontenerami serwerowymi `Application`, `Session` i `Items`. Zademonstrować ich użycie w kodzie po stronie serwera za pomocą opakowań w *pseudosingletony*. Dostęp do którego kontenera musi być dodatkowo chroniony w akcesorze dostępu (`get`) i dlaczego?

6. (1p) Do czego służy plik `app_offline.htm`?