

Projektowanie aplikacji ADO.NET + ASP.NET

Zestaw 6

Moduły i handlers HTTP, ClickOnce

04-12-2012

Liczba punktów do zdobycia: **10/55**

Zestaw ważny do: 08-01-2013

1. **(1p)** Zapoznać się z dowolną implementacją techniki przepisywania URL w ramach ASP.NET. Lista niektórych dostępnych implementacji znajduje się tu: <http://msdn2.microsoft.com/en-us/library/ms972974.aspx>. Zademonstrować jej użycie na własnych przykładach.

Uwaga! Lektura artykułu "URL Rewriting in ASP.NET" Scotta Mitchella jest obowiązkowa. Jako drugi w kolejności można przestudiować artykuł "Rewrite.NET" Roberta Chatteriera.

2. **(2p)** Napisać własny moduł autentykacji, który informację o tożsamości użytkownika przechowa w dowolnym zasobie trwałym po stronie serwera (czy tym zasobem może być kontener sesji i dlaczego?).

Podobnie jak `FormsAuthenticationModule`, w przypadku braku autoryzacji do aktualnego żądania, moduł powinien przekierowywać żądanie do jakiejś strony logowania. W celu sprawdzenia autoryzacji na poziomie zasobu, użyj metody `UrlAuthorizationModule.CheckUrlAccessForPrincipal`.

3. **(1p)** Przygotować solution złożone z projektu aplikacji Web i aplikacji ClickOnce. Deployować aplikację ClickOnce na serwer aplikacji do witryny aplikacji Web i pozwolić użytkownikowi uruchomić aplikację ClickOnce z poziomu aplikacji Web przez odpowiednio spreparowany link do zasobu `*.application`.
4. **(1p)** Rozszerzyć poprzednie zadanie o usługę aplikacyjną (Web Service) przy pomocy którego aplikacja ClickOnce będzie odwoływać się do logiki na serwerze aplikacji. Wygenerować klasę proxy po stronie aplikacji ClickOnce i pokazać w jaki sposób dynamicznie konfigurować adres zwrotny usługi aplikacyjnej tak, żeby nie trzeba było rekonfigurować aplikacji ClickOnce ręcznie tylko żeby adres usługi był konstruowany dynamicznie na podstawie lokalizacji z której pobrana została aplikacja ClickOnce.
5. **(2p)** Rozszerzyć poprzednie zadanie o współdzielenie tożsamości aplikacji Web i usługi aplikacyjnej - obie powinny być zabezpieczone tym samym ciastkiem Forms Authentication. Na wykładzie omówiono sposób współdzielenia tożsamości oparty o jawnie wywoływaną metodę logowania, natomiast tu w zadaniu zrealizować wariant oparty o automatyczne przekazanie wartości ciastka Forms w parametrach aplikacji.

Wskazówka! <http://netpl.blogspot.com/2008/02/clickonce-webservice-and-shared-forms.html>

Pytanie dodatkowe: jak radzić sobie z wygasaniem ciastka Forms Authentication w przypadku bezczynności użytkownika? Zwykła aplikacja Web przekieruje sterowanie do strony logowania, a co powinna zrobić aplikacja ClickOnce?

6. **(1p)** Przygotować własną usługę aplikacyjną (ASP.NET WebService lub WCF) i klienta do niej korzystającego bezpośrednio z protokołu HTTP przez `TcpClient` lub `Socket`.

Wskazówka. <http://netpl.blogspot.com/2007/05/klient-webservices.html>.

7. **(1p)** Przygotować usługę ASP.NET WebService (*.asmx) o określonym kontrakcie oraz usługę WCF (*.svc) o tym samym kontrakcie i wiązaniu `BasicHttpBinding`.

Pokazać, że oba typy klas proxy (*stare* i *nowe*, czyli te dziedziczące z `SoapHttpClientProtocol` i te dziedziczące z `ClientBase`), mogą odwoływać się zamiennie do obu serwisów. Inaczej - skonstruować oba typy klas proxy i połączyć się każdą z nich do obu serwisów i odczytywać dane.

8. **(1p)** Do usługi z poprzedniego zadania wygenerować proxy za pomocą dowolnej innej technologii wytwarzania oprogramowania (Java, PHP, ...?) i pokazać że WSDL/SOAP rzeczywiście określa dobrą platformę interoperacyjności - składanie usług wykonanych w różnych technologiach jest oczywiste.

Wiktor Zychla