

Projektowanie aplikacji ADO.NET + ASP.NET

Zestaw 4

Autentykacja, autoryzacja

12-11-2013

Liczba punktów do zdobycia: **12/42**

Zestaw ważny do: 10-12-2013

1. (**1p**) Zaprezentuj w praktyce przedstawiany na wykładzie mechanizm autentykacji **Windows**. Ściślej - przygotuj aplikację, w której użytkownik zostanie rozpoznany jako aktualnie zalogowany użytkownik systemu operacyjnego. Pokaż, że potrafisz sterować dostępem do poszczególnych zasobów aplikacji za pomocą mechanizmu autoryzacji (użytkownicy przypisani do odpowiednich grup zabezpieczeń mają lub nie dostęp do wybranych stron).

Naucz się różnicy między autentykacją w trybie podstawowym (przeglądarka zapyta o login i hasło) od uwierzytelniania w trybie **Windows** (użytkownik systemu/domeny zostanie rozpoznany automatycznie).

2. (**2p**) Zaimplementuj i użyj we własnej aplikacji takiego dostawcę usługi uwierzytelniania (**MembershipProvider**), który potwierdzi tożsamość użytkownika w bazie danych Microsoft SQL Server, w tabeli **USER**, w której zapisana będzie nazwa i email użytkownika, oraz tabeli **PASSWORD** gdzie zapisane będą skojarzone z użytkownikiem: skrót hasła, sól (salt), liczba rund haszowania oraz data ustawienia hasła (zgodnie ze schematem przedstawionym na wykładzie).

Zbuduj formularz dodawania/rejestracji użytkownika, który po utworzeniu konta poprawnie zapisze dane do obu tabel.

Pokaż że użytkownicy poprawnie logują się do aplikacji.

3. (**1p**) Poprzednie zadanie rozwiń o implementację usługi dostawy ról (**RoleProvider**), gdzie role zapisane byłyby w tabeli **ROLES**, a powiązanie wiele-do-wielu użytkowników z rolami w tabeli **USERSROLES**.

Dostęp do zasobów można zabezpieczyć przez wskazanie ról użytkowników którzy mogliby do tych ról mieć dostęp. Pokaż, że można to robić zarówno dla pojedynczych zasobów (sekcja **location** w **web.config**) oraz całych podfolderów (osobny, zdegenerowany **web.config**).

4. (**2p**) Jak korzystać z informacji o rolach użytkowników w aplikacji?

Pokaż, że potrafisz zablokować dostęp do podglądu i edycji **wybranego** elementu **GridView/ListView** dla użytkowników będących w konkretnej roli.

Na przykład pole **PESEL** powinien widzieć każdy, a edytować tylko użytkownik będący w roli **ADMINISTRATOR**, zaś pole **PENSJA** powinien widzieć i edytować tylko użytkownik w roli **PLACOWA**.

5. (1p) Pokaż, że potrafisz posługiwać się sekcją `UserData` ciastka Forms. Ściślej - napisz takiego dostawcę usługi informowania o rolach, który listę ról użytkownika zapamięta w sekcji `UserData` ciastka Forms w momencie logowania, a przy każdym żądaniu dostarczenia listy ról będzie wydobywał je z tej sekcji.

Zadanie to ma ma celu oswojenie się ze strukturą ciastka forms oraz interfejsem, który pozwala na jego tworzenie oraz na dostęp do informacji w nim zawartych (klasa `FormsAuthenticationTicket`).

6. (1p) Użyj wbudowanego w szablon witryny ASP.NET w VS2012/3 dostawcy usługi autentykacji protokołu OAuth2 (`OpenAuth`) do zbudowania witryny umożliwiającej zalogowanie się użytkownika za pomocą dwóch wybranych dostawców autentykacji (Facebook/Google/Microsoft/Twitter/itd.). Naucz się rejestrować swoją aplikację u co najmniej jednego dostawcy, tam gdzie to jest konieczne.
7. (2p) Naucz się korzystać z biblioteki `DotNetOpenAuth` do autentykacji OAuth2 do dwóch wybranych dostawców autentykacji.
8. (2p) W automatycznie tworzonym szablonie witryny, zewnętrzni dostawcy wykorzystywani są do tworzenia kont użytkowników w domyślnej bazie danych.

Zrefaktoryzuj poprzedni przykład tak, żeby nie tyle pozwalać kontom użytkowników tworzyć się po zautentykowaniu się u zewnętrznego dostawcy, co raczej - wiązać użytkowników logujących się u zewnętrznego dostawcy z tymi zapisanymi w lokalnej, własnej bazie danych użytkowników z jednego z poprzednich zadań wtedy i tylko wtedy, kiedy zgadzają się ich adresy e-mail.

Wiktor Zychła