

Opis projektu zaliczeniowego na Programowanie Aplikacji Sieciowych

Serwer uploadu/downloadu plików – MiniCloud

Michał Korniak

Klienci wysyłają do serwera pliki w dowolnym formacie, mają możliwość pobrania z serwera listy wysłanych plików (każdy użytkownik ma własne konto na serwerze służące do przechowywania plików), pobrania dowolnego wcześniej wrzuconego pliku, klient może również pobrać pliki innych użytkowników, jeśli są one mu udostępnione.

Projekt został napisany w języku C#. Aby umożliwić zaimplementowanie mechanizmu autentykacji oraz autoryzacji do zasobów wykorzystano bazę SqlServer. Same zasoby są przechowywane na dysku. Oprócz serwera stworzono również dwa programy klienckie. Pierwszy z nich pracuje w konsoli umożliwiając samodzielne pisanie komend (problemem w tym przypadku jest wysłanie plików, gdyż wewnątrz komendy musi się znaleźć base64 takiego pliku). Drugi program kliencki jest aplikacją okienkową stworzoną za pomocą frameworka WPF, wykorzystuje ona wszystkie możliwości serwera w sposób przyjazny użytkownikowi.

Klient komunikuje się z serwerem za pomocą niżej opisanych komend. Aby zapewnić integralność danych na początku każdego requestu i każdego response'a znajduje się suma kontrolna MD5 właściwej części wiadomości. Po odebraniu takiego response'a lub requesta ponownie generujemy sumę kontrolną i porównujemy ją z otrzymaną.

We właściwej części każdego requesta znajdują się informacje o wykorzystywanym kontrolerze oraz metodzie wewnątrz kontrolera, dalej podawane są parametry jeśli takowe istnieją. Cały interfejs serwera został wygenerowany automatycznie poprzez mechanizm korzystający z refleksji języka C#. Przykładowo, aby stworzyć kontroler o nazwie Test, zawierający metodę MetodaTestowa, wystarczy stworzyć klasę Test implementującą interfejs IController, a wewnątrz niej metodę o takiej nazwie. Aby potem wywołać tą metodę z poziomu klienta wystarczy wysłać requesta o treści „test metoda_testowa”. Mechanizm ten zawiera zabezpieczenia przed wysyłaniem niekompletnych komend (bez wymaganych parametrów).

Serwer obsługuje następujące komendy:

<i>Komenda</i>	<i>Opis</i>	<i>Wymagane zalogowanie</i>
<i>account register <user_name> <password></i>	<i>rejestracja</i>	<i>NIE</i>
<i>account login <user_name> <password></i>	<i>logowanie</i>	<i>NIE</i>
<i>account say_my_name</i>	<i>wypisanie nazwy aktualnie zalogowanego użytkownika</i>	<i>TAK</i>
<i>directory create <path> <directory_name></i>	<i>stworzenie nowego katalogu wewnątrz ścieżki</i>	<i>TAK</i>
<i>directory remove <path></i>	<i>usunięcie katalogu</i>	<i>TAK</i>
<i>directory structure</i>	<i>wypisanie struktury katalogów (razem z udostępnionymi)</i>	<i>TAK</i>
<i>directory share <donee_name> <path></i>	<i>udostępnienie katalogu danemu użytkownikowi</i>	<i>TAK</i>
<i>directory stop_share <donee_name> <path></i>	<i>zatrzymanie udostępniania katalogu dla danego użytkownika</i>	<i>TAK</i>
<i>directory list_users_with_access <path></i>	<i>wypisanie użytkowników uprawnionych do przeglądania danego katalogu</i>	<i>TAK</i>
<i>file download <owner_name> <file_path></i>	<i>jeśli zalogowany użytkownik ma dostęp do pliku następuje jego pobieranie w base64</i>	<i>TAK</i>
<i>file remove <file_path></i>	<i>usuwanie pliku z własnego katalogu użytkownika</i>	<i>TAK</i>
<i>file upload <directory_path> <file_name> <base64_file></i>	<i>wysyłanie pliku (base64) na serwer</i>	<i>TAK</i>