

Zaawansowane Techniki Programowania Java

Laboratorium #7

- Java API for XML Web Services -

Program: Rozszerz funkcjonalność programu *Ewidencja pracowników* o komunikację przy użyciu SOAP-Based Web Services (*JAX-WS, Java API for XML Web Services*).

Zastosuj:

- SOAP-based WebService serwer - **uruchamiany przy starcie aplikacji wątek-demon** udostępniający wszystkie dane o pracownikach klientowi sieciowemu łączącemu się za pośrednictwem protokołu SOAP. Zapewnij jednoczesną obsługę wielu klientów SOAP (Multithreading the Endpoint Publisher). Udostępnienie danych o pracownikach wymaga posłużenia się kluczem autoryzującym (tokenem) w postaci base64 uzyskanym przy pozytywnym wyniku autentykacji za pośrednictwem zdalnego obiektu RMI opracowanym w ramach laboratorium *Remote Method Invocation (java.rmi)*.
- SOAP-based WebService klient – łączący się za pośrednictwem protokołu SOAP klient sieciowy (jako wybór w menu aplikacji) pobierający ze zdalnej aplikacji wszystkie dane o pracownikach i posługujący się przy wywołaniu usługi kluczem autoryzującym (tokenem) w postaci base64 uzyskanym przy pozytywnym wyniku autentykacji za pośrednictwem zdalnego obiektu RMI opracowanym w ramach laboratorium *Remote Method Invocation (java.rmi)*.
- Marshalling do postaci XML (w części serwera) i unmarshalling z XML (w części klienta) obiektów przesyłanych za pośrednictwem ustanowionego połączenia Web Service przy wykorzystaniu rozwiązania opracowanego w ramach laboratorium *Java Architecture for XML Binding (javax.xml.bind)*.
- Implementując rozwiązanie należy wykorzystać czynnościowy wzorzec projektowy Strategia (https://en.wikipedia.org/wiki/Strategy_pattern) umożliwiający aplikacji (w roli klienta) wymienne stosowanie sposobu (strategii) komunikacji z serwerem w zależności od wyboru funkcjonalności opracowanej w ramach bieżącego laboratorium *Java API for XML Web Services (javax.xml.ws)* lub w ramach laboratorium *Networking with Sockets (java.net)*.

Literatura:

1. Kalin, M.: Java WebServices: Up and Running, O'REILLY, 2013.
2. Plik zewnętrzny: #07 – ZTPJ_JAX-WS.pdf.

Polecenie: Zaprojektuj i zaimplementuj rozwiązanie dla następującego problemu:



- 1) Widok formatki menu głównego programu:

MENU

1. Lista pracowników
2. Dodaj pracownika
3. Usuń pracownika
4. Kopia zapasowa
5. Pobierz dane z sieci

Wybór>

2) Widok ekranu Pobierz dane z sieci:

5.1.Pobierz dane z sieci		
Podaj użytkownika	:	jkowalski
Podaj hasło	:	*****
Protokół [T]CP/IP czy [S]OAP?	:	s
Adres	:	192.168.0.2
Port	:	1111
Zasób	:	ewid
Autentykacja użytkownika...Sukces		
Ustanawianie połączenia... Sukces		
Pobieranie... Sukces		
Czy zapisać pobrane dane? [T]/[N] t		
Zapisywanie... Sukces		

Uwaga! Kody źródłowe programu po sprawdzeniu na zajęciach laboratoryjnych muszą zostać połączone w jeden plik źródłowy i przesłany na adres kkaska@wi.zut.edu.pl:

1) plik z kodem źródłowym musi mieć nazwę:

numer_indeksu.ztpj.lab07.java

np.: 666.ztpj.lab07.java.

W przypadku dużych plików można użyć kompresji ZIP lub GZ. Wówczas plik archiwum z kodem źródłowym musi mieć nazwę:

numer_indeksu.ztpj.lab07.java.(zip|gz)

2) plik musi zostać wysłany z konta poczty uczelni (zut.edu.pl),

3) nagłówek maila musi mieć postać:

ZTPJ I2 XXY LAB07

gdzie XXY to numer grupy (np. ZTPJ I2 14A LAB07),

4) w pierwszych trzech liniach plików z kodem źródłowym w komentarzach jednoliniowych musi znaleźć się:

- informacja identyczna z zamieszczoną w nagłówku maila,
- imię i nazwisko wysyłającego oraz
- adres email, z którego wysłał wiadomość,

5) email nie powinien zawierać żadnej treści (tylko załącznik).

Dostarczone kody źródłowe będą analizowane pod kątem wykrywania plagiatów. Nie wysłanie wiadomości lub wysłanie jej w formie niezgodnej z powyższymi wymaganiami będzie traktowane jako brak programu i skutkowało brakiem oceny z niniejszej pracy.