
Laborprotokoll

Web Services in JAVA

**Systemtechnik Labor
5BHITT 2015/16, Gruppe Y**

Simon Wortha

Note:

Betreuer: Prof. Borko

Version 1

Begonnen am 11. März 2016

Beendet am 11. März 2016

Inhaltsverzeichnis

1	Einführung	3
1.1	Ziele	Fehler! Textmarke nicht definiert.
1.2	Voraussetzungen	Fehler! Textmarke nicht definiert.
1.3	Aufgabenstellung	Fehler! Textmarke nicht definiert.
2	Ergebnisse	4
2.1	Asymmetrische Verschlüsselung ..	Fehler! Textmarke nicht definiert.
2.2	Symmetrische Verschlüsselung ..	Fehler! Textmarke nicht definiert.
2.3	Die Kommunikation in Java	Fehler! Textmarke nicht definiert.
2.4	Naming Service	Fehler! Textmarke nicht definiert.

1 Einführung

Diese Übung zeigt die Anwendung von mobilen Diensten in Java.

1.1 1.1 Ziele

Das Ziel dieser Übung ist eine Webanbindung zur Benutzeranmeldung in Java umzusetzen. Dabei soll sich ein Benutzer registrieren und am System anmelden können.

Die Kommunikation zwischen Client und Service soll mit Hilfe von JAX-RS (Gruppe1) umgesetzt werden.

1.2 1.2 Voraussetzungen

- Grundlagen Java und Java EE
- Verständnis über relationale Datenbanken und dessen Anbindung mittels JDBC oder ORM-Frameworks
- Verständnis von Restful Webservices

1.3 1.3 Aufgabenstellung

Es ist ein Webservice mit Java zu implementieren, welches eine einfache Benutzerverwaltung implementiert. Dabei soll die Webapplikation mit den Endpunkten /register und /login erreichbar sein.

Registrierung

Diese soll mit einem Namen, einer eMail-Adresse als BenutzerID und einem Passwort erfolgen. Dabei soll noch auf keine besonderen Sicherheitsmerkmale Wert gelegt werden. Bei einer erfolgreichen Registrierung (alle Elemente entsprechend eingegeben) wird der Benutzer in eine Datenbanktabelle abgelegt.

Login

Der Benutzer soll sich mit seiner ID und seinem Passwort entsprechend authentifizieren können. Bei einem erfolgreichen Login soll eine einfache Willkommensnachricht angezeigt werden.

Die erfolgreiche Implementierung soll mit entsprechenden Testfällen (Acceptance-Tests bez. aller funktionaler Anforderungen mittels JUnit)

dokumentiert werden. Es muss noch keine grafische Oberfläche implementiert werden! Verwenden Sie auf jeden Fall ein gängiges Build-Management-Tool (z.B. Maven). Dabei ist zu beachten, dass ein einfaches Deployment möglich ist (auch Datenbank mit z.B. file-based DBMS).

2 Ergebnisse

2.1 Durchführung

Die Kommunikation zwischen Client und Service funktioniert mit JAX-RS, was mit einem Modul für Spring implementiert wurde.

Bei der Umsetzung habe ich mich von folgenden Repository inserieren lassen:

<https://github.com/pkalauner-tgm/dezsys09-java-webservices>

3 Probleme

Bei den heruntergeladenen Dependencies beim Gradle Projekt, kam es durch das TGM-Internet zu starken Verzögerungen

4 Quellen

"Android Restful Webservice Tutorial – Introduction to RESTful webservice – Part 1"; Posted By Android Guru on May 1, 2014; online:

<http://programmerguru.com/android-tutorial/android-restful-webservice-tutorial-part-1/>

"REST with Java (JAX-RS) using Jersey - Tutorial"; Lars Vogel; Version 2.5; 15.12.2015; online: <http://www.vogella.com/tutorials/REST/article.html>

"O Java EE 7 Application Servers, Where Art Thou? Learn all about the state of Java EE app servers, a rundown of various Java EE servers, and benchmarking."; by Antonio Goncalves; Java Zone; Feb. 10, 2016; online: <https://dzone.com/articles/o-java-ee-7-application-servers-where-art-thou>

"Heroku makes it easy to deploy and scale Java apps in the cloud"; online: <https://www.heroku.com/>