JUST DO IT
HÖHERE ABTEILUNG FÜR INFORMATIONSTECHNOLOGIE

tgm
Die Schule der Technik

Laborprotokoll

Web Services in Java

Systemtechnik Labor
5BHITT 2015/16, Gruppe Y

Christoph Hackenberger

Version 1.0

Note:

Betreuer: M. Borko

Begonnen am 12. Februar 2016

Beendet am 18 Februar 2016

Inhaltsverzeichnis

1 Einführung

Diese Übung zeigt die Anwendung von mobilen Diensten in Java..

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 Voraussetzungen

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

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 dokumentiert werden. Es muss noch keine grafische Oberfläche implementiert werden! Verwenden Sie auf jeden Fall ein gängiges Build-Management-Tool

1.4 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/

2 Aufsetzen des Application Servers

Zuerst muss ein Application Server aufgesetzt werden. Für diese Übung habe ich mich für den Apache Tomcat (Version 9) entschieden. Tomcat wird nach dem Tutorial[1] aufgesetzt.

Nach der Installation kann der Tomcat mit folgendem Befehl gestartet werden:

```
sudo /bin/su - <tomcat-user> -c /usr/share/tomcat/bin/startup.sh
```

Zum Herunterfahren wird folgender Befehl verwendet:

```
sudo /bin/su - <tomcat-user> -c /usr/share/tomcat/bin/shutdown.sh
```

Nun müssen noch die Zugänge für das Deployment eingerichtet werden dazu in die Datei */usr/share/tomcat/conf/tomcat-user.xml* folgende Einträge bearbeiten:

<role rolename="admin-gui"/>
<user username="chackenberger" password="password" roles="admin-gui, manager-gui"/>

<role rolename="admin-script"/>
<user username="deploy" password="password" roles="admin-script, manager-script"/>

User mit der Rolle admin-gui und manager-gui haben Zugriff auf die Weboberfläche.
User mit der Rolle admin-script und manager-script haben Zugang auf die CLI von tomcat. Diese wird zum Deployment mittels zb. dem tomcat maven plugin verwendet.

Nun muss der Tomcat Server neugestartet werden.

3 Projekt und Deployment einrichten

Als erstes haben wir die Projekt Vorlage[2] in ein Maven Projekt umgewandelt.

Wir mussten die Libraries welche mit der Vorlage mitkommen (unter WebContent/WEB-INF/lib) aus dem Projekt entfernen da wir diese über Maven Dependencies laden und es sonst zu Konflikten kommt.

Für die entsprechenden Dependencies siehe bitte (pom.xml)

Da wir nicht wie in der Vorlage Jersey 1.X sonder 2.X verwenden muss auch die web.xml in WebContent/WEB-INF entsprechen angepasst werden:

<servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>

und

<param-name>jersey.config.server.provider.packages</param-name>

Nun muss das Deployment eingerichtet werden. Daher müssen wir zuerst einmal mittels des maven-war-plugin ein entsprechendes .war Archiv erzeugen. Hier muss auch angeben wo das Directory mit den Web-Resources liegt siehe pom.xml

Für das eigentliche Deployment verwenden wir das tomcat7-maven-plugin. Bei diesem muss die URL zur tomcat CLI angeben werden, der Pfad unterwelchem die Applikation später laufen soll und die vorher in der tomcat-user.xml konfigurierten Zugangsdaten.

4 Persistierung

4.1 MySQL

Die Projekt Vorlage[2] verwendet eine MySQL DB zur Persistierung der User-Daten.

Der Zugang zur Datenbank wird in der Klasse Constants konfiguriert.

Hier einfach den entsprechenden DB-Namen, JDBC-Connection String, Zugangsdaten angeben werden.

Außerdem muss in der Datenbank noch folgende Tabelle[2] erstellt werden:

```
CREATE TABLE IF NOT EXISTS `user` (
  `name` varchar(50) NOT NULL,
  `username` varchar(50) NOT NULL,
  `password` varchar(50) NOT NULL,
  `register_dt` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`username`)
)
```

4.2 SQLite

Als zweite Option habe ich eine SQLite Datenbank zur Verfügung gestellt diese wird nun auch in der Constants Klasse konfiguriert.

Dafür einfach den Class-String und JDBC-Connection String von MySQL auskommentieren und jeweils für SQLite einkommentieren. Die Zugangsdaten sind in dem Fall dann egal.

Achtung: Als dbName muss der absolute Pfad zum SQLite File angeben werden (am Server), da sonst ein neues File (in dem die benötigte Tabelle nicht existiert) innerhalb des Working-Dir des Application Servers erzeugt wird.

Zu beachten bei Verwendung von SQLite werden (aufgrund anderer Error Codes als bei MySQL) keine detaillierten Fehlermeldungen wie zb. User existiert bereits zurückgegeben.

5 AcceptanceTests

5.1 Neuen User Registrieren

name: pgrs
username:abc
password:xyz

Request: register/doregister?name=pqrs&username=abc&password=xyz

Response: {"tag":"register","status":true}

TEST OK

5.2 Doppelten User Registrieren

name: pgrs
username:abc
password:xyz

Request: register/doregister?name=pqrs&username=abc&password=xyz

Response: {"tag":"register","status":false,"error_msg":"You are already registered"}

TEST OK

5.3 Fehlender GET-Parameter

name: test
password: blub

Request: register/doregister?name=test&password=blub

Response: {"tag":"register","status":false,"error_msg":"Error occured"}

TEST OK

5.4 Erfolgreicher Login

username: abc
password: xyz

Request: login/dologin?username=abc&password=xyz

Response: {"tag":"login","status":true}

TEST OK

5.5 Falsches Passwort

username: abc
password: passwort

Request: login/dologin?username=abc&password=passwort

Response: {"tag":"login","status":false,"error_msg":"Incorrect Email or Password"}

TEST OK

5.6 Nicht existierender Benutzer

username: zwergbumsti
password: passwort

Request: login/dologin?username=zwergbumsti&password=passwort

Response: {"tag":"login","status":false,"error_msg":"Incorrect Email or Password"}

TEST OK

5.7 Passwort Parameter fehlt bei Login

username: abc

Request: login/dologin?username=abc

Response: {"tag":"login","status":false,"error_msg":"Incorrect Email or Password"}

TEST OK

5.8 Username Paramenter fehlt bei Login

assword: xyz

Request: login/dologin?password=xyz

Response: {"tag":"login","status":false,"error_msg":"Incorrect Email or Password"}

TEST OK

6 GitHub Repo

https://github.com/chackenberger/Useraccount

7 Quellen

[1] Installing Java 8.x and Tomcat 8.x on Debian Jessie (and RedHat too), Wolf Paulus' Journal, verfügbar unter: https://wolfpaulus.com/journal/software/tomcat-jessie/ [abgerufen am 12.2.2016]

[2] Android Restful Webservice Tutorial – How to create RESTful webservice in Java – Part 2, Programmer Guru, verfügbar unter, http://programmerguru.com/android-tutorial/android-restful-webservice-tutorial-how-to-create-restful-webservice-in-java-part-2/ [abgerufen am 12.2.2016]