
GK9_3

Cloud-Datenmanagement

Systemtechniklabor
5BHIT 2017/18

Johannes Bishara

Note:
Betreuer: Michael Borko

Version 1.1
Begonnen am 5. Oktober 2017
Beendet am 12. Oktober 2017

Inhaltsverzeichnis

1	Einführung	1
1.1	Ziele	1
1.2	Vorraussetzungen	1
1.3	Aufgabenstellung	1
2	Ergebnisse	2
2.1	Tutorial	2
2.1.1	Programm erstellen	2
2.1.2	pom.xml	2
2.2	Register	3

1 Einführung

Diese Übung zeigt die Anwendung von mobilen Diensten.

1.1 Ziele

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

Die Kommunikation zwischen Client und Service soll mit Hilfe einer REST Schnittstelle umgesetzt werden.

1.2 Voraussetzungen

- Grundlagen einer höheren Programmiersprache
- Verständnis über relationale Datenbanken und dessen Anbindung mittels ODBC oder ORM-Frameworks
- Verständnis von Restful Webservices

1.3 Aufgabenstellung

Es ist ein Webservice 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 Unit-Tests) dokumentiert werden. Verwenden Sie auf jeden Fall ein gängiges Build-Management-Tool (z.B. Maven oder Gradle). Dabei ist zu beachten, dass ein einfaches Deployment möglich ist (auch Datenbank mit z.B. file-based DBMS).

2 Ergebnisse

2.1 Tutorial

Zu Beginn habe ich das Tutorial mit Jersey und JAX-RS umgesetzt, um eine Basiskonfiguration zu erhalten, auf die ich aufbauen kann. Folgende Schritte waren notwendig:

2.1.1 Programm erstellen

Zuerst erstellt man ein 'Dynamic Web Project'. Dabei wählt man als Target Runtime Apache Tomcat v9.0. Da bei der Erstellung des Projekts kein web.xml File vorhanden war, habe ich einen Deployment Descriptor Stub generieren lassen.

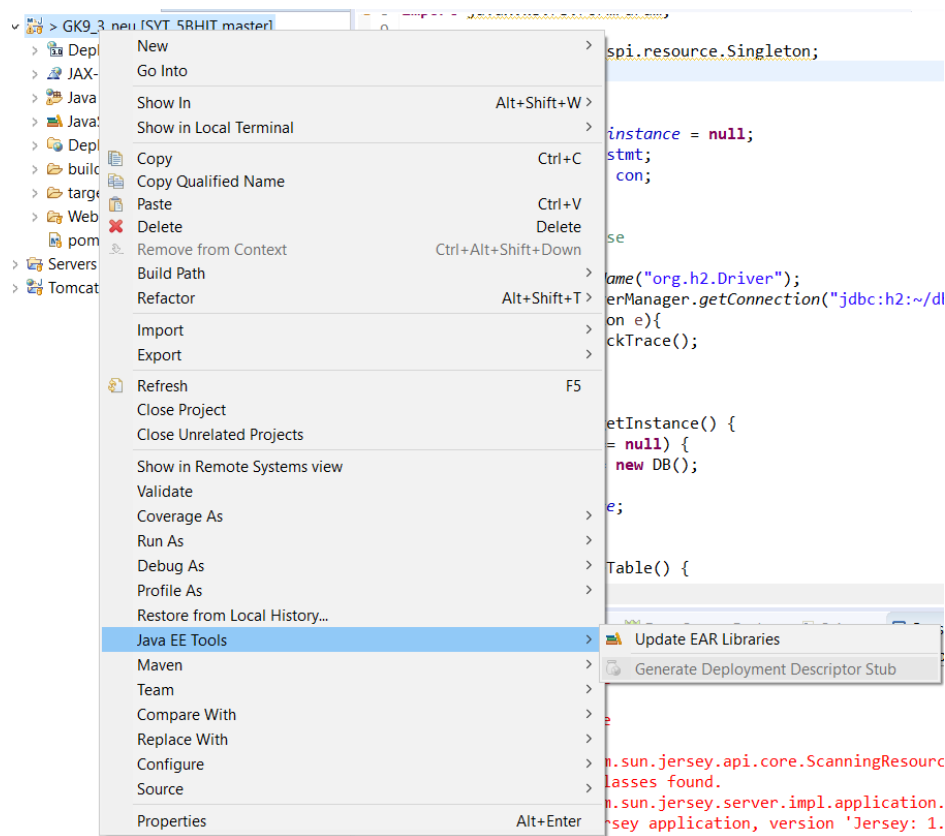


Abbildung 1: Generate Deployment Descriptor Stub

2.1.2 pom.xml

Nun konvertiert man das Projekt in ein Maven-Projekt und added folgende Dependencies in das pom.xml File:

- asm.jar
- jersey-bundle.jar

- json.jar
- jersey-server.jar

Hier ein Ausschnitt wie man eine Dependencie in ein pom.xml hinzufügt.

```
<dependency>
  <groupId>com.sun.jersey</groupId>
  <artifactId>jersey-bundle</artifactId>
  <version>1.19.4</version>
</dependency>
```

Abbildung 2: Dependency-Struktur in pom.xml

2.2 Register

Beim Register habe ich als Path /register gemäß der Aufgabenstellung angegeben.

```
@Path("/register")
public class Register {
```

Abbildung 3: Path von Register

2.2.1 register() Methode

In der Klasse habe ich die Methode register() geschrieben, welche das Registrierungsformular zur Verfügung stellt. Wichtig ist dabei die @Produces Annotation. Diese sagt aus, dass die Methode etwas produziert/darstellt. Der Parameter besagt was genau dabei erstellt wird, in diesem Fall eine HTML-Seite. Die Klasse liefert ein String zurück, indem das HTML für die Seite dabeisteht.

```
@GET
@Produces(MediaType.TEXT_HTML)
public String register() throws Exception {
```

Abbildung 4: Annotations von register()

```
return "<html>\r\n" +
    "<head><meta charset=\"UTF-8\"></head>\r\n" +
    "<body>\r\n" +
    "<form method=\"post\">\r\n" +
    "  First name:<br>\r\n" +
    "  <input type=\"text\" name=\"firstname\"><br>\r\n" +
    "  Last name:<br>\r\n" +
    "  <input type=\"text\" name=\"lastname\"><br>\r\n" +
    "  E-Mail:<br>\r\n" +
    "  <input type=\"text\" name=\"mail\"><br>\r\n" +
    "  Password:<br>\r\n" +
    "  <input type=\"password\" name=\"pw\">\r\n" +
    "  <input type=\"submit\" value=\"Submit\">\r\n" +
    "</form>\r\n" +
    "</body>\r\n" +
    "</html>\r\n" +
    "";
```

Abbildung 5: HTML von Register

Tabellenverzeichnis

Listings

Abbildungsverzeichnis

1	Generate Deployment Descriptor Stup	2
2	Dependency-Struktur in pom.xml	3
3	Path von Register	3
4	Annotations von register()	3
5	HTML von Register	4