

---

# Laborprotokoll

## Distributed Computing GK9.3

---

Systemtechnik Labor  
5BHIT 2017/18

Matthias Weiss

Note:  
Betreuer: M. Borko

Version 0.1  
Begonnen am 5. Oktober 2017  
Beendet am 5. Oktober 2017

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Ziele . . . . .	1
1.2	Voraussetzungen . . . . .	1
1.3	Aufgabenstellung . . . . .	1
1.4	Bewertung . . . . .	2
1.5	Quellen . . . . .	2
<b>2</b>	<b>Ergebnisse</b>	<b>3</b>
2.1	Tabelle . . . . .	3
2.2	Aufzählung . . . . .	3
2.3	Code . . . . .	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).

## 1.4 Bewertung

- Gruppengröße: 1 Person
- Anforderungen "überwiegend erfüllt"
  - Dokumentation und Beschreibung der angewendeten Schnittstelle
  - Aufsetzen einer Webservice-Schnittstelle
  - Registrierung von Benutzern mit entsprechender Persistierung
- Anforderungen für Gänze erfüllt"
  - Login und Rückgabe einer Willkommensnachricht
  - Überprüfung der funktionalen Anforderungen mittels Regressionstests

## 1.5 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 2.7; 27.09.2017; online: <http://www.vogella.com/tutorials/REST/article.html>

„Django REST framework“; Tom Christie; online: <http://www.django-rest-framework.org/>

„Eve. The Simple Way to REST“; Nicola Iarocci; online: <http://python-eve.org/>

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

## 2 Ergebnisse

passwort muss mit sha512 encoded werden -> abgleichen

## Literatur

- [1] A.S. Tanenbaum and M. Van Steen. *Verteilte Systeme: Prinzipien und Paradigmen*. Pearson Studium. Addison Wesley Verlag, 2007.

## Tabellenverzeichnis

1	Lorem ipsum dolor sit amet [1] . . . . .	3
---	--	---

## Listings

1	Implizite Transaktion [1] . . . . .	3
---	-------------------------------------	---

## Abbildungsverzeichnis