JUST DO IT
HÖHERE ABTEILUNG FÜR
INFORMATIONSTECHNOLOGIE

tgm
Die Schule der Technik

Laborprotokoll

Dezentrale Systeme - Web Services in Java

Systemtechnik Labor
5BHITT 2015/16, Gruppe B

Thomas Taschner

Version 1.0
Begonnen am 11. März 2016
Beendet am 18. März 2016

Note:
Betreuer: Michael Borko

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 + 2) 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 Datebanktabelle 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).

Bewertung: 16 Punkte

- Aufsetzen einer Webservice-Schnittstelle (4 Punkte)

  Registrierung von Benutzern mit entsprechender Persistierung (4 Punkte)
- Login und Rückgabe einer Willkommensnachricht (3 Punkte)
- AcceptanceTests (3 Punkte)
- Protokoll (2 Punkte)

2 Quellen

[1] [2] [3] [4]

3 Ergebnisse

Für die Umsetzung dieser Übung wurden folgende Technologien verwendet: Spring-Framework, da es im Vergleich zu anderen Frameworks dieser Art einfach zu verwenden und aufzusetzen ist. Ebenso von Vorteil ist der integrierte Applikationsserver. Die Verwendung von JAX-RS wurde im Rahmen dieser Übung vorgeschrieben. Unser Betreuer gab uns den Hinweis H2, eine dateibasierte und hier leichter zu verwendete Datenbank, statt einer relationalen Datenbank, wie beispielsweise MySQL, zu verwenden. Als Buildmanagement-Tool wurde Maven verwendet, welches bisher wunderbar seine Dienste verrichten konnte. Ein in den Quellen verlinktes Tutorial ('Bootiful' Java EE Support in Spring Boot 1.2) wurde übernommen und entsprechend an die in dieser Übung benötigte Funktionalität angepasst. Wichtige, noch nicht vorhandene Teile waren beispielsweise die RESTful Endpoints für dem Login und die Registrierung, aber auch die Datenbankanbindung mussten angepasst werden. [5, 6, 7, 8, 9]

3.1 Link zum Repository

Die Abgabe wurde, wie üblich, auf GitHub zur Verfügung gestellt. [10]

4 Akzeptanzkriterien

4.1 Registrierung

Die Überprüfung der Funktionalität gemäß den vorgeschriebenen Anforderungen wurde mittels des Firefox-Plugins *HttpRequester* durchgeführt. [11]

4.1.1 Erfolgreiche Registrierung

Sollten alle Parameter gegeben sein, so wird ein Account erstellt.

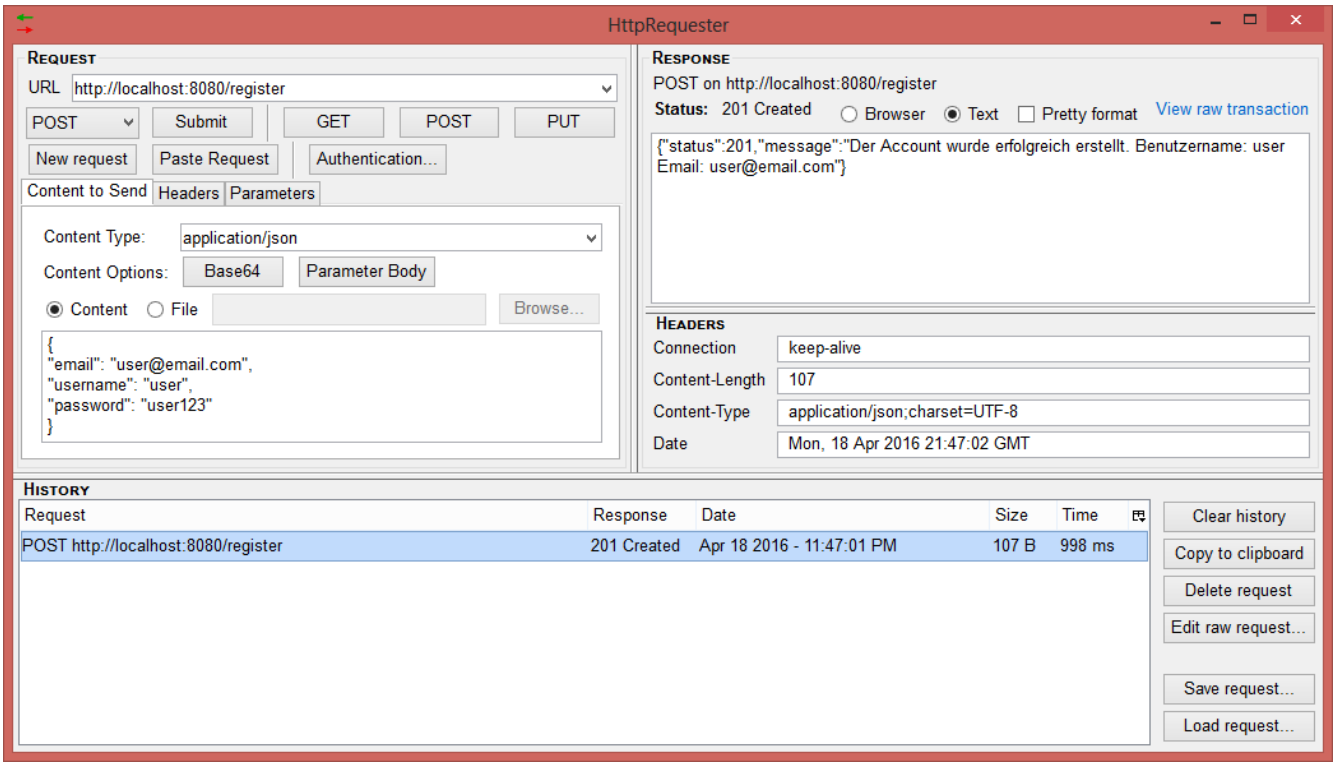

Abbildung 1: Erfolgreiche Registrierung des Benutzers

4.1.2 Fehlende(r) Parameter

Sollte ein oder mehrere Parameter fehlen, so wird der Account nicht erstellt. Der Benutzer wird auf seinen Fehler entsprechend hingewiesen.

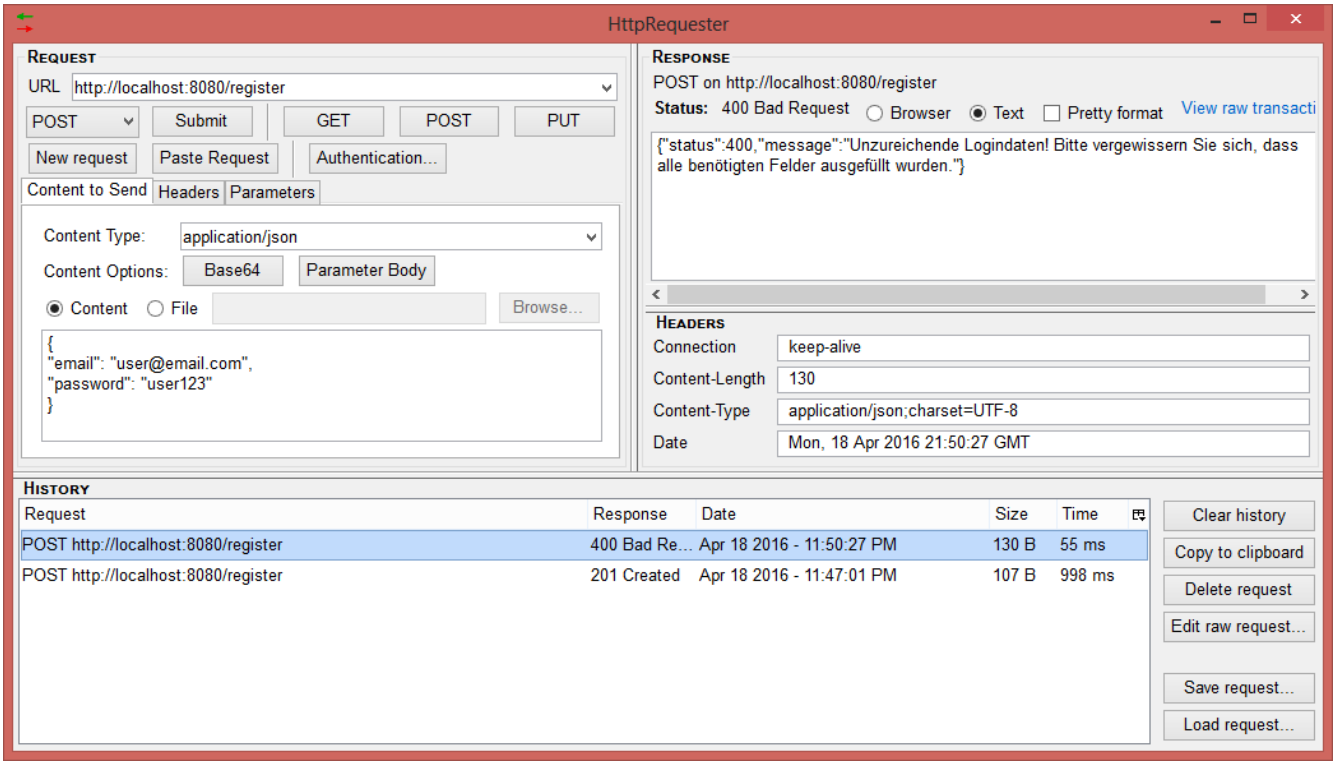

Abbildung 2: Eingabe von fehlerhaften Parametern

4.1.3 Bereits vorhandener Benutzer

Sollte der Benutzer bereits in der Datenbank registriert sein, so wird der Account nicht angelegt. Der Benutzer wird darüber entsprechend informiert.

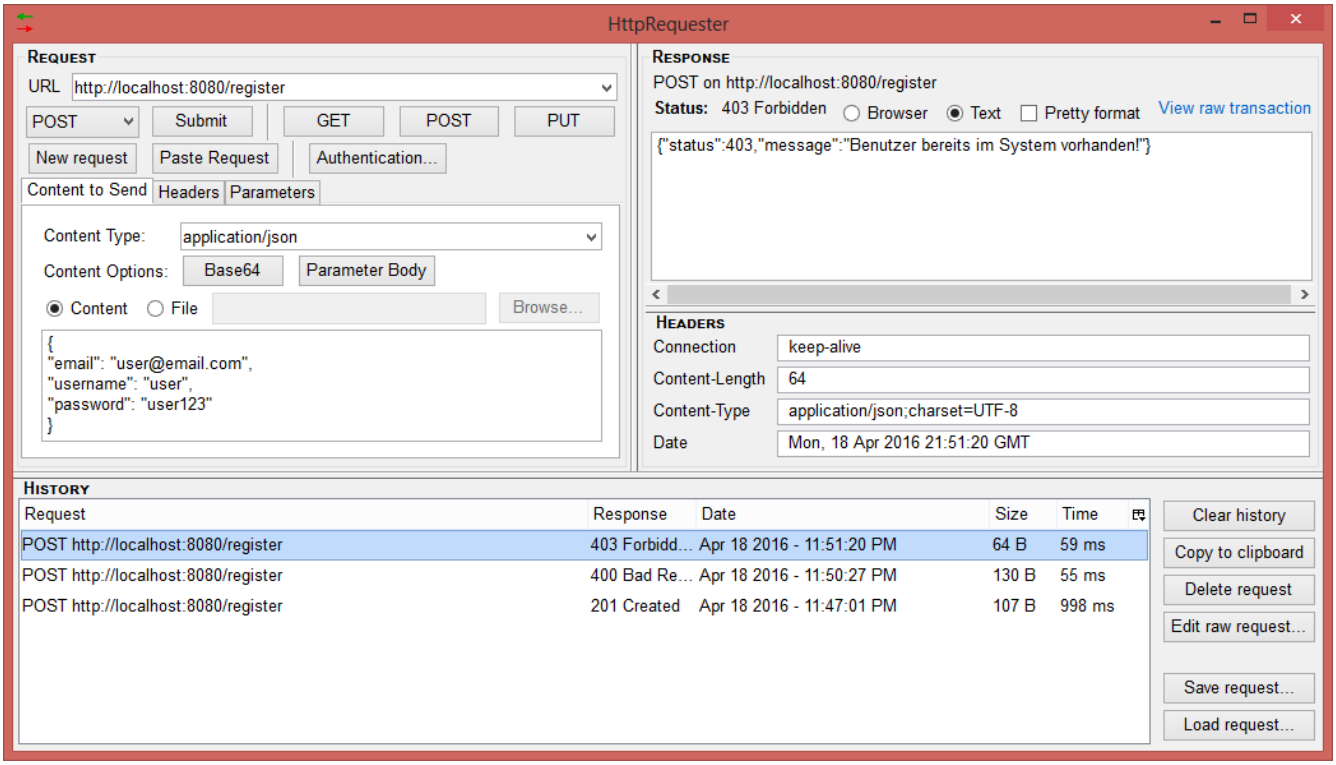

Abbildung 3: Benachrichtigung über bereits vorhandenen Account

4.2 Login

4.2.1 Erfolgreicher Login

Sollten alle Parameter gegeben und der Benutzer in der Datenbank vorhanden sein, so wird dieser mit Erfolg angemeldet.

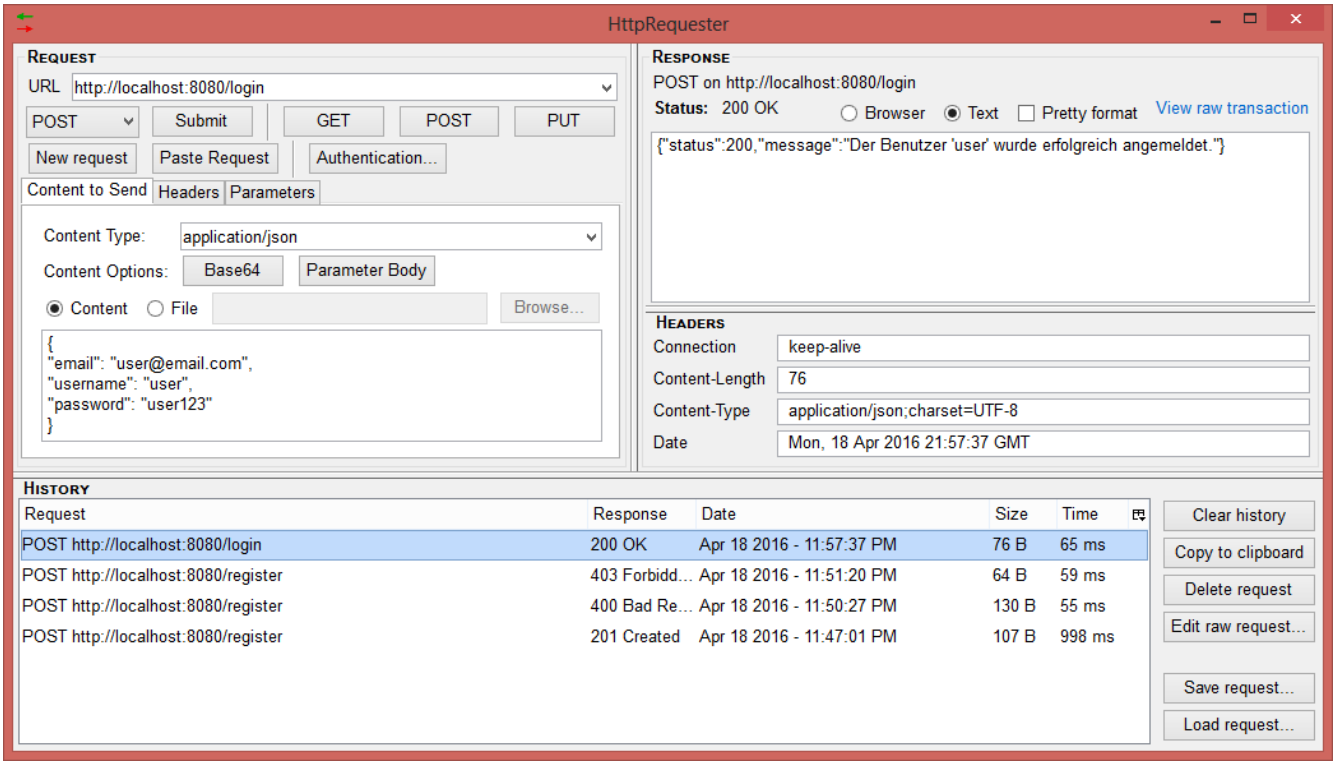

Abbildung 4: Erfolgreicher Login eines Benutzers

4.2.2 Fehlende(r) Parameter

Sollte ein oder mehrere Parameter beim Loginvorgang fehlen, so wird der Benutzer entsprechend darauf hingewiesen.

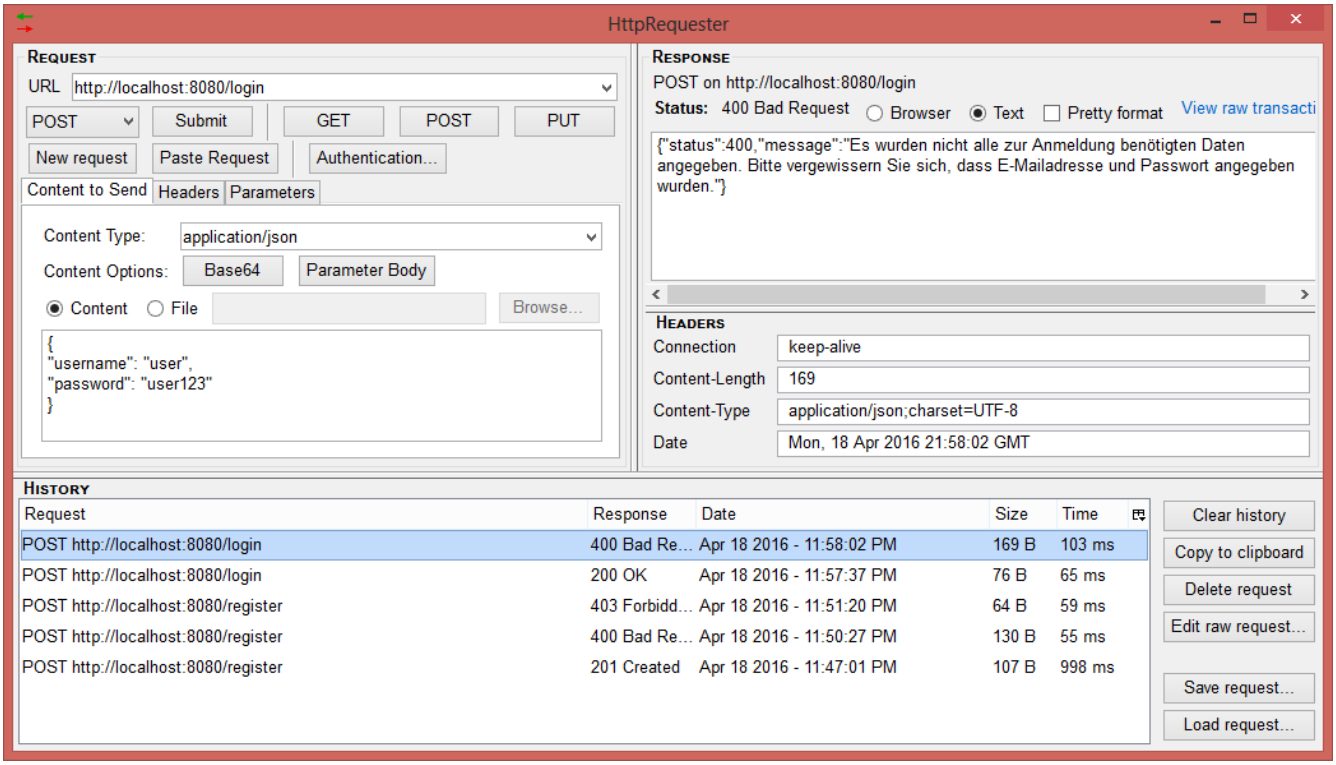

Abbildung 5: Benachrichtigung über fehlende Parameter

4.2.3 Benutzer nicht vorhanden

Sollte der Benutzer nicht in der Datenbank gefunden werden können, so wird dieser entsprechend darauf hingewiesen.

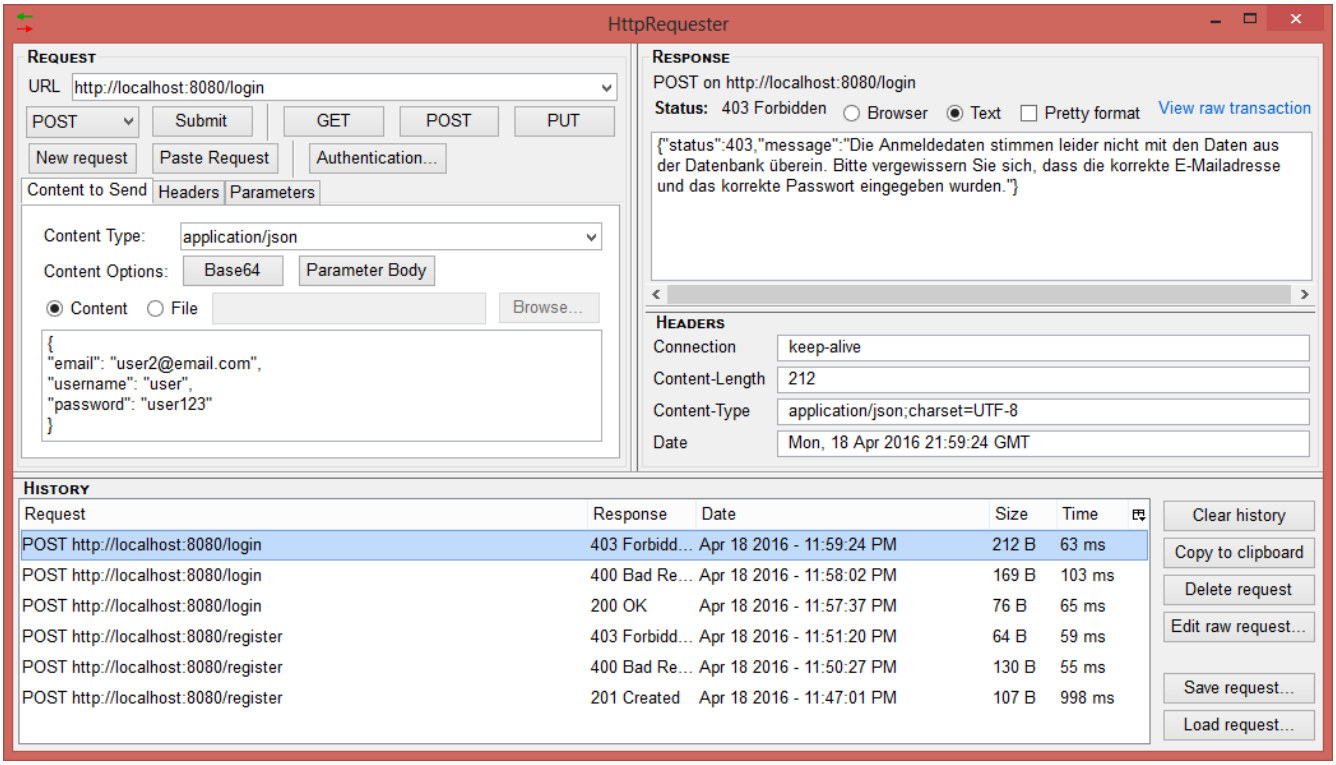

Abbildung 6: Nicht vorhandener Benutzer

4.2.4 Probleme

- Die meisten Probleme hatte ich dadurch das TGM-Netz mal wieder recht langsam war und ich warten musste bis alle Dependencies heruntergeladen waren.

- Es war mir nicht möglich eine HTTP Error Message (also ein HTTP Status- Code 20x) ohne jeglichen Content zu senden, dann bekam der „Client“ immer eine 404 Message.

- Manchmal kam es zu komischen Dependencies Problemen, wenn man zu viele oder fehlende Abhängigkeiten hatte.

4.3 Fazit

Zunächst gab es Probleme mit dem Deployment, welche durch einen Blick auf bestehende Lösungen von anderen Mitschülern und einer kurzen Recherche mithilfe einer Suchmaschine innerhalb von ca. 15 Minuten gelöst werden konnten. Die benötigte Zeit zum Lösen der Aufgabe lag bei ca. 5 Stunden.

Literatur

[1] Android restful webservice tutorial – introduction to restful webservice – part 1. http://programmerguru.com/android-tutorial/android-restful-webservice-tutorial-part-1/. Zuletzt besucht: 11.03.2016.

[2] Rest with java (jax-rs) using jersey - tutorial. http://www.vogella.com/tutorials/REST/article.html. Zuletzt besucht: 11.03.2016.

[3] 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. https://dzone.com/articles/o-java-ee-7-application-servers-where-art-thou. Zuletzt besucht: 11.03.2016.

[4] Heroku makes it easy to deploy and scale java apps in the cloud. https://www.heroku.com/. Zuletzt besucht: 11.03.2016.

[5] 'bootiful' java ee support in spring boot 1.2. http://spring.io/blog/2014/11/23/bootiful-java-ee-support-in-spring-boot-1-2. zuletzt besucht: 11.03.2016.

[6] Spring. http://spring.io/. zuletzt besucht: 11.03.2016.

[7] Jax-rs. http://docs.oracle.com/javaee/6/tutorial/doc/giepu.html. zuletzt besucht: 11.03.2016.

[8] Maven. https://maven.apache.org/. zuletzt besucht: 11.03.2016.

[9] H2 database engine. http://www.h2database.com/html/main.html. zuletzt besucht: 11.03.2016.

[10] tgm-ttaschner's dezsys09 repository. https://github.com/tgm-ttaschner/DezSys09. Zuletzt besucht: 11.03.2016.

[11] Httprequester. https://addons.mozilla.org/de/firefox/addon/httprequester/. zuletzt besucht: 11.03.2016.

Abbildungsverzeichnis