

DezSys07

SOA and RESTful Webservice

Dezentrale Systeme

5BHIT 2015/16

Geyer Stefan, Ritter Mathias

Version 1.0

Betreuer: Prof. Micheler Begonnen am 4. Dezember 2015

Note: Beendet am 17. Dezmber 2015

Inhaltsverzeichnis

[Einführung 3](#_Toc431895375)

[Ziele 3](#_Toc431895376)

[Voraussetzungen 3](#_Toc431895377)

[Aufgabenstellung 3](#_Toc431895378)

[Ergebnisse 4](#_Toc431895379)

# Einführung

Dieses Protokoll-Template soll helfen den Laborübungsteil entsprechend dokumentieren zu können.

## Ziele

Hier werden die zu erwerbenden Kompetenzen und deren Deskriptoren beschrieben. Diese werden von den unterweisenden Lehrkräften vorgestellt.

Dies kann natürlich auch durch eine Aufzählung erfolgen.

## Voraussetzungen

Welche Informationen sind notwendig um die Laborübung reibungslos durchführen zu können? Hier werden alle Requirements der Lehrkraft detailliert beschrieben und mit Quellen untermauert.

Hier zum Beispiel die Architektur der Common Object-Request-Broker Architecture:

## Aufgabenstellung

Hier wird dann die konkrete Aufgabenstellung der Laborübung definiert.

Nun kommt ein Seitenumbruch, um eine klare Trennung der Schülerarbeit zu bestimmen.

# Ergebnisse

### Datenmodell & Generierung der Testdaten

Als Datenmodell wurde ein JSON-Objekt definiert, welches ein Attribut "name" und ein Attribut "description" enthält. Diese werden in einer dokumentenorientierten NoSQL Datenbank, MongoDB, abgelegt.

Um eine Million Testdaten zu generieren, wurde der Datengenerator der Seite "GenerateData" verwendet. [1] Dieser muss allerdings heruntergeladen werden, da online maximal 100 Datensätze generiert werden können.

Nach dem Herunterladen benötigt man einen funktionsfähigen Webserver, PHP & MySQL Setup, um das Installationsscript auszuführen. Im ersten Schritt gibt man die Daten der Datenbank an. Falls noch kein User und keine Datenbank erstellt wurde, muss diese vorher noch erstellt werden. Danach baut das Script eine Datenbankverbindung auf. Nun werden die Testdaten in die Datenbank geladen. Im letzten Schritt gibt man Username & Passwort zum Aufruf der Seite an. Danach kann man den DataGenerator wie bei der Online-Version verwenden.

Um die oben erwähnten Daten zu generieren, gibt man nun bei der ersten Spalte als Bezeichnung "name" ein. Als Datentyp wählt man Name. In die Optionsspalte schreibt man "Name Name Surname". In der zweiten Spalte gibt man als Bezeichnung "description" ein und wählt "random number of words" aus. In der Optionsspalte stellt man zwischen 20 und 30 Wörter ein. Zuletzt stellt man 100.000 Datensätze (Maximum) und JSON (Simle) ein. Nun klickt man auf Generate und nach ein paar Sekunden lädt das generierte JSON-File herunter. Diesen Vorgang wiederholt man 10 Mal und fügt danach alle Daten zu einer JSON-Datei zusammen.

## Provisionierung & Deployment

Um die Webapplikation und den SOA-Client auszuführen, muss zuerst eine virtuelle Maschine mittels Vagrant aufgesetzt werden. Dazu muss man VirtualBox und Vagrant herunterladen und installieren. [3, 4]

Literaturverzeichnis

[1] Generatedata Daten-Generator, Version 3.2.4 [Online]

Verfügbar unter: <http://www.generatedata.com>

[zuletzt abgerufen am 17.12.2015]

[3] Vagrant, Development environments made easy [Online]

Verfügbar unter: <https://www.vagrantup.com>

[zuletzt abgerufen am 17.12.2015]

[4] VirtualBox, Powerful x86 and AMD64/Intel64 virtualization [Online]

Verfügbar unter: <https://www.virtualbox.org>

[zuletzt abgerufen am 17.12.2015]