April 19, 2015

Wolfgang Mair, Andreas Vogt

TGM 5AHITT

SOA-REST Webservice

Implementierung eines Service Oriented Architecture und eines Restful Webservices

Table of Contents

[Aufgabenstellung 2](#_Toc417241355)

[Aufwandsschätzung 3](#_Toc417241356)

[Arbeitszeit 3](#_Toc417241357)

[Technologiebeschreibung 4](#_Toc417241358)

[REST 4](#_Toc417241359)

[SOA 4](#_Toc417241360)

[WSDL 4](#_Toc417241361)

[SOAP 4](#_Toc417241362)

[UDDI 4](#_Toc417241363)

[Hibernate 4](#_Toc417241364)

[JDBC 5](#_Toc417241365)

[Spring 5](#_Toc417241366)

[Erarbeitete Informationen 5](#_Toc417241367)

[Testing 6](#_Toc417241368)

[Rest 6](#_Toc417241369)

[Weboberfläche 7](#_Toc417241370)

[SOA 9](#_Toc417241371)

[Quellen 11](#_Toc417241372)

# Aufgabenstellung

**DezSys08 - Service Oriented Architecture and RESTful Webservice**

Das neu eröffnete Unternehmen **iKnow Systems** ist spezialisiert auf **Knowledgemanagement** und bietet seinen Kunden die Möglichkeiten Daten und Informationen jeglicher Art in eine Wissensbasis einzupflegen und anschließend in der zentralen Wissensbasis nach Informationen zu suchen (ähnlich wikipedia).

Folgendes ist im Rahmen der Aufgabenstellung verlangt:

* Entwerfen Sie ein Datenmodell, um die Eintraege der Wissensbasis zu speichern und um ein optimitiertes Suchen von Eintraegen zu gewaehrleisten. **[2Pkt]**
* Entwickeln Sie mittels RESTful Webservices eine Schnittstelle, um die Wissensbasis zu verwalten. Es muessen folgende Operationen angeboten werden:  
  - **Hinzufuegen** eines neuen Eintrags  
  - **Aendern** eines bestehenden Eintrags  
  - **Loeschen** eines bestehenden Eintrags  
  Alle Operationen muessen ein Ergebnis der Operation zurueckliefern. **[3Pkt]**
* Entwickeln Sie in **Java** ein **SOA Webservice**, dass die Funktionalitaet **Suchen** anbietet und das **SOAP** Protokoll einbindet. Erzeugen Sie fuer dieses Webservice auch eine **WSDL**-Datei. **[3Pkt]**
* Entwerfen Sie eine **Weboberflaeche**, um die **RESTful Webservices** zu verwenden. **[3Pkt]**
* Implementieren Sie einen **einfachen Client** mit einem User Interface (auch Commandline UI moeglich), der das **SOA Webservice** aufruft. **[2Pkt]**
* Dokumentieren Sie im weiteren Verlauf den Datentransfer mit SOAP. **[1Pkt]**
* Protokoll ist erforderlich! **[2Pkt]**

**Info:**  
Gruppengroesse: 2 Mitglieder  
Punkte: 16

Zum Testen bereiten Sie eine Routine vor, um die Wissensbasis mit einer **1 Million Datensaetze** zu fuellen. Die Datensaetze sollen mindestens eine Laenge beim Suchbegriff von 10 Zeichen und bei der Beschreibung von 100 Zeichen haben! **Ist die Performance bei der Suche noch gegeben?**

**Links:**  
***JEE Webservices:***   
<http://docs.oracle.com/javaee/6/tutorial/doc/gijti.html>

***Apache Web Services Project:***   
<http://ws.apache.org/>

***Apache Axis/Axis2:***<http://axis.apache.org>

<http://axis.apache.org/axis2/java/core/>  
***IBM Article: Java Web services - JAXB and JAX-WS in Axis2:***<http://www.ibm.com/developerworks/java/library/j-jws8/index.html>

# Aufwandsschätzung

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Aufgabe | Geschätzt (Stunden) | Ist (Stunden) |
| Andreas | Rest Schnittstelle | 2 | 4 |
| Wolfgang | Rest Schnittstelle | 4 | - |
| Andreas | Datenmodell | 2 | 1 |
| Wolfgang | Datenmodell | 3.5 | - |
| Andreas | Rest Oberfläche | 4 | 5 |
| Wolfgang | Rest Oberfläche | 5 | - |
| Andreas | SOAP Search | 5.5 | - |
| Wolfgang | SOAP Search | 5.5 | 7 |
| Andreas | Client | 2.5 | - |
| Wolfgang | Client | 2 | 3 |
| Andreas | Doku | 1.5 | - |
| Wolfgang | Doku | 0.5 | 1 |
| Andreas | Protokoll | 2 | 2 |
| Wolfgang | Protokoll | 1 | 1 |

# Arbeitszeit

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Aufgabe | Datum | Ist (Stunden) |
| Andreas | Rest Schnittstelle | 15.04.2015 |  |
| Andreas | Datenmodell | 15.04.2015 |  |
| Wolfgang | Datenmodell | 15.04.2015 |  |
| Andreas | Rest Oberfläche | 15.04.2015 |  |
| Wolfgang | SOAP Search | 19.04.2015 | 7 |
| Wolfgang | Client | 16.04.2015 | 3 |
| Andreas | Doku | 15.04.2015 |  |
| Wolfgang | Doku | 19.04.2015 | 1 |
| Andreas | Protokoll | 15.04.2015 |  |
| Wolfgang | Protokoll | 19.04.2015 | 1 |

|  |  |  |
| --- | --- | --- |
| Name | Geschätzt (Gesamt) | Ist (Gesamt) |
| Andreas | 19.5 | 12 |
| Wolfgang | 21.5 | 12 |

# Technologiebeschreibung

## REST

REST basiert auf Prinzipien, die in der größten verteilten Anwendung eingesetzt werden - dem World Wide Web. Das World Wide Web stellt selbst eine gigantische REST Anwendung dar. Viele Suchmaschinen, Shops oder Buchungssysteme sind ohne Absicht bereits als REST basierter Web Services verfügbar. Die *REpresentational State Transfer Architektur* ist ein Architektur Modell, welches beschreibt, wie das Web funktionieren sollte. Das Modell soll als Anleitung und als Referenz für zukünftiger Erweiterungen dienen. REST ist kein Produkt oder Standard. REST beschreibt, wie Web Standards in einer Web gerechten Weise einsetzt werden können.[8]

## SOA

Vereinfacht kann SOA als Methode bzw. Paradigma angesehen werden, die vorhandenen EDV-Komponenten wie Datenbanken, Server und Websites in Dienste zu kapseln und dann so zu koordinieren („Orchestrierung“), dass ihre Leistungen zu höheren Diensten zusammengefasst und anderen Organisationsabteilungen oder Kunden zur Verfügung gestellt werden können. Maßgeblich sind also nicht technische Einzelaufgaben wie Datenbankabfragen, Berechnungen und Datenaufbereitungen, sondern die Zusammenführung dieser IT-Leistungen zu „höheren Zwecken“ – wie Ausführen einer Bestellung oder Prüfen der Rentabilität einer Abteilung usw. –, die eine Organisationsabteilung anbietet. [7]

## WSDL

Die Web Services Description Language ist eine plattform-, programmiersprachen- und protokollunabhängige Beschreibungssprache für Netzwerkdienste zum Austausch von Nachrichten auf Basis von XML. WSDL ist ein industrieller Standard des World Wide Web Consortium.[6]

## SOAP

SOAP ist ein Netzwerkprotokoll, mit dessen Hilfe Daten zwischen Systemen ausgetauscht und Remote Procedure Calls durchgeführt werden können. SOAP ist ein industrieller Standard des World Wide Web Consortium. [4]

## UDDI

## Hibernate

**Hibernate** (englisch für *Winterschlaf halten*) ist ein Open-Source-Persistenz- und ORM-Framework für Java. Für .NET ist eine portierte Version namens **NHibernate** verfügbar.

Hibernates Hauptaufgabe ist das *Object-Relational Mapping* (O-R-Mapping, kurz ORM). Dies ermöglicht es, gewöhnliche Objekte mitAttributen und Methoden (im Java-Bereich POJOs genannt) inrelationalen Datenbanken zu speichern und aus entsprechenden Datensätzen wiederum Objekte zu erzeugen. Beziehungen zwischen Objekten werden auf entsprechende Datenbank-Relationen abgebildet.

Darüber hinaus bietet Hibernate Mechanismen zur Kompatibilität mit verschiedenen Datenbanken. Die zum Datenbankzugriff erforderlichen SQL-Statements werden nicht explizit in SQL programmiert, sondern von Hibernate in Abhängigkeit vom SQL-Dialekt der verwendeten Datenbank generiert. [3]

## JDBC

Java Database Connectivity ist eine Datenbankschnittstelle der Java-Plattform, die eine einheitliche Schnittstelle zu Datenbanken verschiedener Hersteller bietet und speziell auf relationale Datenbanken ausgerichtet ist. [5]

## Spring

Das **Spring Framework** (kurz *Spring*) ist ein quelloffenes Framework für die Java-Plattform. Ziel des Spring Frameworks ist es, die Entwicklung mit Java/Java EE zu vereinfachen und gute Programmierpraktiken zu fördern. Spring bietet mit einem breiten Spektrum an Funktionalität eine ganzheitliche Lösung zur Entwicklung von Anwendungen und deren Geschäftslogiken; dabei steht die Entkopplung der Applikationskomponenten im Vordergrund.

# Erarbeitete Informationen

**Einträge genereieren:**

Um die Last bei den generieren der Einträge zu verkleinern ist es klüger die Inserts in größere Stücke zu trennen. Bei zu großen Einträgen auf einmal kann es zur stätigen Verlangsamung des Schreibens kommen, oder sogar zu groß für einen einzigen Dateneintrag sein.

# Probleme

Die Restapplikation konnte nicht zu einem Ausführbaren Jar gemacht werden da Spring das Zeichen „@“ erkennt welches in Hibernate verwendet wird. Das Program kann in IntelliJ ausgeführt werden nachdem die Dependencies aus dem Mavenfile pom.xml geladen wurden. Weiters lässt sich die Weboberfläche nur mit einem Webserver ausführen.

# Testing

## Rest

Die Restful Applikation wurde mittels Spring entwickelt.

Der Port lässt sich im File :

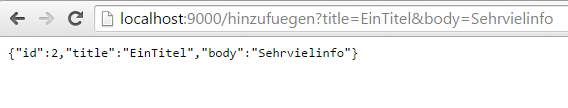
application.properties ändern.

Der Server läuft in meinem Fall local auf Port 9000 und die Funktionen lassen sich wie folgt aufrufen:

**Hinzufügen eines neuen Eintrags:**

<http://localhost:9000/hinzufuegen?title=test&body=Sehrvielinfo>

title ist der Titel des neuen Eintrags und body der Inhalt.



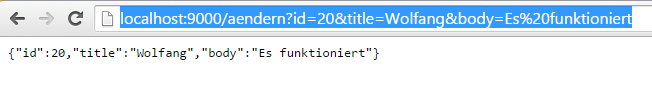


Die ID die im JSON File angezeigt wird wird beim Serverneustart zurückgesetzt deshalb auch die verschiedenen IDS.

**Eintrag Ändern:**

Hier muss die Id , der Titel und der Body angegeben werden

<http://localhost:9000/aendern?id=20&title=Wolfang&body=Es%20funktioniert>



Before

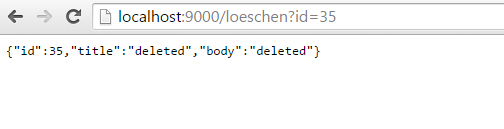


After



**Eintrag loeschen**

Um einen Eintrag zu löschen muss man nur die ID angeben.



Before:

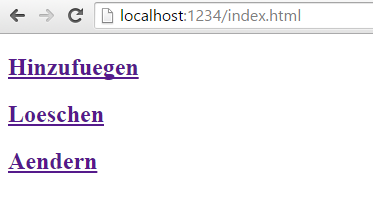


After:

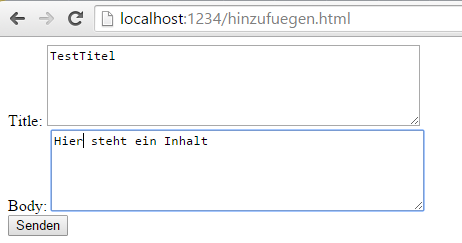


## Weboberfläche

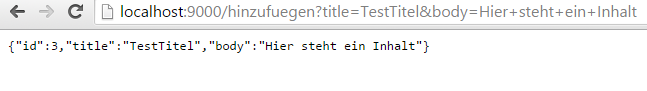
Die Weboberfläche wurde in html erstellt und man muss nur die Daten in Textfelder angeben und den bestätigugnsbutton drücken. Als Webserver wurde eine Vagrantmaschine mit integrierten Webserver verwendet:



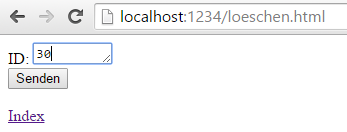
**Hinzufuegen**

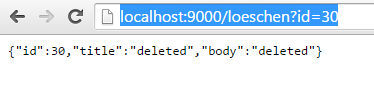


Beim drücken des Senden buttons wird nur der Pfad der Rest Methode mit den Parametern aus den Textboxen aufgerufen

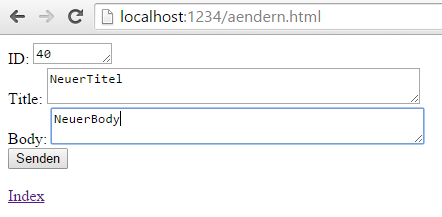


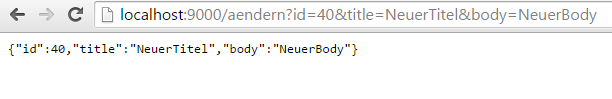
**Loeschen**





**Aendern**





## SOA

Starten des Publishers und starten des Clients danach.

**Funktioniert** - beide sind aktiv

Eingabe in den Client nach einem gesuchten Wort.

**Funktioniert** - Nach dem generieren der verlangten 1 Million Einträgen gibt er aus wie viele der Einträge dem gesuchten Wort ähneln.

Eingabe in den Client ohne generieren der Einträge.

**Fehlgeschlagen** - Ist nicht möglich

Eingabe eines leeren Feldes (ENTER ohne Eingabe) lässt die Eingabe wiederholen.

**Funktioniert** - Eine erneute Eingabe wird verlangt

Eingabe eines Wertes das nicht vorkommt lässt die Einträge generieren und gibt zurück das nichts gefunden wurde.

**Mäßig** - Einträge werden weiterhin generiert, gibt aber aus das nichts gefunden wurde.

# Quellen

[1] <https://spring.io/guides/gs/producing-web-service/>

[2] <http://www.mkyong.com/webservices/jax-ws/jax-ws-hello-world-example/>

[3] http://de.wikipedia.org/wiki/Hibernate\_%28Framework%29

[4] http://de.wikipedia.org/wiki/SOAP

[5] [http://de.wikipedia.org/wiki/JDBC](http://www.google.at/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCMQkA4oADAA&url=http%3A%2F%2Fde.wikipedia.org%2Fwiki%2FJDBC&ei=_-8zVcycJ4OjygOE44CQAg&usg=AFQjCNHd1yAwhbfZW_6MSUrbKqRw_I-Pzw&sig2=Uy6vqxhBTeMgFsP1f8_5OQ)

[6] [http://de.wikipedia.org/wiki/WSDL](http://www.google.at/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCAQkA4oADAA&url=http%3A%2F%2Fde.wikipedia.org%2Fwiki%2FWSDL&ei=RvAzVasu5qrLA-LBgJAP&usg=AFQjCNHM8KUF44tPpEhmxVH1fpxU5bvxyw&sig2=Sa_Hvyhjl0eGRsD7FVluWw)

[7] http://de.wikipedia.org/wiki/Serviceorientierte\_Architektur

[8] <http://www.oio.de/public/xml/rest-webservices.htm>

[9] <http://de.wikipedia.org/wiki/Spring_%28Framework%29>