
DezSys-Protokoll

DezSys-07 "SOA and RESTful Webservice"

Dezentrale Systeme
5BHITT 2015/16

Thomas Stedronsky und Sebastian Steinkellner

Version 1.0

Note:

Betreuer: Th. Micheler, M. Borko

Begonnen am 11. Dezember 2015

Beendet 07. Jänner 2016

Inhaltsverzeichnis

1	Einführung.....	3
1.1	Ziele.....	3
1.2	Voraussetzungen.....	3
1.3	Aufgabenstellung.....	3
2	Ergebnisse.....	4
2.1	GUI.....	4
2.2	Probleme bzw. Anmerkungen.....	5
	Request Mapping.....	5
	DB-Connection.....	6
3	Quellen.....	7
4	Abbildungsverzeichnis.....	8
5	Codeverzeichnis.....	9

1 Einführung

Das neu eröffnete Unternehmen iKnow Systems ist spezialisiert auf Knowledge Management 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).

1.1 Ziele

Das Verständnis von REST bzw. SOAP Schnittstellen und deren Verbindung mit einer Datenbank.

1.2 Voraussetzungen

Es muss eine Testdatenbank mit einem Key Value Paar vorhanden sein, diese sollte sinnvolle Daten enthalten. Zum Testen bereiten Sie eine Routine vor, um die Wissensbasis mit einer 1 Million Datensätze zu füllen. Die Datensätze sollen mindestens eine Länge beim Suchbegriff von 10 Zeichen und bei der Beschreibung von 100 Zeichen haben! Ist die Performance bei der Suche noch gegeben?

1.3 Aufgabenstellung

- Entwerfen Sie ein Datenmodell, um die Einträge der Wissensbasis zu speichern und um ein optimiertes Suchen von Einträgen zu gewährleisten. [2Pkt]
- Entwickeln Sie mittels RESTful Webservices eine Schnittstelle, um die Wissensbasis zu verwalten. Es müssen folgende Operationen angeboten werden:
 - Hinzufügen eines neuen Eintrags
 - Ändern eines bestehenden Eintrags
 - Löschen eines bestehenden Eintrags
- Alle Operationen müssen ein Ergebnis der Operation zurückliefern. [3Pkt]
- Entwickeln Sie in Java ein SOA Webservice, das die Funktionalität Suchen anbietet und das SOAP Protokoll einbindet. Erzeugen Sie für dieses Webservice auch eine WSDL-Datei. [3Pkt]
- Entwerfen Sie eine Weboberfläche, um die RESTful Webservices zu verwenden. [3Pkt]
- Implementieren Sie einen einfachen Client mit einem User Interface (auch Commandline UI möglich), der das SOA Webservice aufruft. [2Pkt]
- Dokumentieren Sie im weiteren Verlauf den Datentransfer mit SOAP. [1Pkt]
- Protokoll ist erforderlich! [2Pkt]

2 Ergebnisse

Zu aller erst wurde ein Rest Framework benötigt, hierbei haben wir uns für Spring entschieden, weil dieses Framework alle erforderlichen Funktionen für unseren Anspruch besitzt.

Nach der Auswahl wurde ein passendes Example gefunden auf dem die Anwendung aufgebaut werden konnte[1].

2.1 GUI

Die Weboberfläche für die Steuerung der Anwendung sieht wie folgt aus:

Eintrag anlegen

Aktionen	Name	Suchbegriffe	Datei
bearbeiten, löschen	Test	test	test.html
bearbeiten, löschen	Lehrer	lehrer, schule, unterricht	lehrer.html
bearbeiten, löschen	Wikipedia	wissen, sammlung, vorbild, weiterleitung	wikipedia.php
bearbeiten, löschen	TGM	schule, unterricht, bauwerk	tgm.html

Abbildung 1 Weboberfläche

Um den Benutzer eine bessere Übersicht zu geben wird farblich markiert, ob dieser Link verfügbar ist. In dem Fall von test.html, wo der Link nicht verfügbar ist wird dieser Rot hinterlegt. Der Link [wikipedia.php](#) ist verfügbar und kann somit angeklickt werden.

Um einen neuen Eintrag einzufügen sieht die GUI wie folgt aus:

Eintrag anlegen

Abbildung 2 Eintrag anlegen

Hier können Name, Suchbegriff sowie der Dateiname angegeben werden.

Außerdem gibt es GUIs für das Bearbeiten bzw. Löschen.

Abbildung 3 Eintrag löschen

Abbildung 4 Eintrag bearbeiten

2.2 Probleme bzw. Anmerkungen

Request Mapping

bei @RequestMapping gibt es 2 Möglichkeiten wie man eine Seite angeben kann.

Version 1:

@RequestMapping("seite") → So wird die Seite direkt angegeben

Version 2:

@RequestMapping(value="seite", method=RequestMethod.x, produces="")

Das x kann bei dieser Methode GET, PUT, DELETE oder auch eine andere Methode sein. Außerdem kann das value unter anderem ein JSON Array sein bzw. ein einfacher String ("seite" oder {"seite1", "seite2"})

Beispiel:

```
@RequestMapping(value="/eintrag",method=RequestMethod.GET,produces="text/plain")
public Eintrag get(@RequestParam(value="titel",defaultValue="test") String
titel) {
    return Eintrag.load(conn.getConnection(), titel);
}
```

Code 1 Request Mapping

Hierbei war das Problem herauszufinden wie man wie man die Request-Methode richtig angibt.

DB-Connection

```
/**
 * loads the database driver and connects to the database. closes the programm,
 * if there is an error
 */
private void connStart() {

    try {
        Class.forName(drivers.get(dbtype)).newInstance();
        System.out.println("Driver " + drivers.get(dbtype) + " successfully
loaded.");
    } catch(Exception ex) {
        System.err.println("Can't find Database driver class! " +
ex.getMessage());
        System.exit(1);
    }

    try {
        connection = (Connection) DriverManager.getConnection("jdbc:" + dbtype +
"://" + host + "/" + database, usercache.getUser(), usercache.getPassword());
        System.out.println("Successfully connected to " + database + " on " +
host);
    } catch(SQLException ex) {
        System.err.println("DB connection error! " + ex.getMessage());
    }
}
```

Code 2 DB-Connection

Die Treiber für die Datenbank werden hier geladen und somit wird die Verbindung zur Datenbank aufgebaut.

3 Quellen

[1] Getting Started · Building a RESTful Web Service, <https://spring.io/guides/gs/rest-service/>, zuletzt besucht am 02.01.16

[2] spring - AbstractTransactionalTestNGSpringContextTests not working with @Transactional multi transaction managers - Stack Overflow, <http://stackoverflow.com/questions/34660912/abstracttransactionaltestngspringcontexttests-not-working-with-transactional-mu>, zuletzt besucht am 22.12.15

[3] java - Spring web-flow object serialization error - Stack Overflow, <http://stackoverflow.com/questions/34658464/spring-web-flow-object-serialization-error>, zuletzt besucht am 02.01.16

[4] Java EE vs Spring. Oder: Was ist eigentlich ein Standard?, <https://blog.codecentric.de/2011/03/java-ee-vs-spring-oder-was-ist-eigentlich-ein-standard/>, zuletzt besucht am 03.01.16

4 Abbildungsverzeichnis

Abbildung 1 Weboberfläche	4
Abbildung 2 Eintrag anlegen	4
Abbildung 3 Eintrag löschen	5
Abbildung 4 Eintrag bearbeiten	5

5 Codeverzeichnis

Code 1 Request Mapping	5
Code 2 DB-Connection	6