

Aufgabe 4 - RESTful Webservice¹ mit Java

Ziel

Das Ziel dieser Aufgabe ist der Erwerb von Kompetenzen in der Entwicklung von RESTful Web Services. Dazu gehört das grundsätzliche Verständnis des [REST](#)-Paradigmas. Darüberhinaus zielt diese Aufgabe auf das Verständnis der Funktionsweise von Protokollen (*HTTP*) auf Ebene 7 (*application layer*) des ISO-OSI-Schichtenmodells ab.

Aufgabe

Die Aufgabe besteht in der Implementierung des *Produktionsplanungssystems (PPS)*, dessen Einbettung in das Gesamtsystem bereits in Aufgabe 3 beschrieben wurde. Zur Implementierung gehören ein Server mit einer in-memory Datenbank und einer HTTP-Schnittstelle nach dem REST-Paradigma (RESTful Webservice), sowie ein Client.

Der **Server** muss folgende Funktionalitäten mittels RESTful Webservice anbieten:

- Alle Fertigungsaufträge zurückgeben
- Alle Fertigungsaufträge sortiert nach ihrem Rang zurückgeben
- Neuen Fertigungsauftrag anlegen, und hinten in der Reihenfolge anhängen
- Neuen Fertigungsauftrag anlegen, und vorne in der Reihenfolge einfügen
- Neuen Fertigungsauftrag anlegen, und an bestimmter Position in der Reihenfolge einfügen
- Maschinenzuordnung eines Fertigungsauftrags ändern
- Rang eines Fertigungsauftrags ändern
- Einen Fertigungsauftrag löschen
- Alle Fertigungsaufträge löschen

Fertigungsaufträge haben eine laufende Nummer als eindeutige Kennnummer, eine Bezeichnung, einen Rang, der die Reihenfolge der Abarbeitung (Priorität) anzeigt, und ein Textfeld für die Bezeichnung der Maschine, an der der Fertigungsauftrag gerade bearbeitet wird. Die Daten sollen in einer ganz einfachen in-memory Datenbank mit nur einer einzigen Tabelle gespeichert werden.

Das HTTP-Protokoll verwendet [Statuscodes](#). Der Server muss je nach Fehler einen passenden Statuscode melden. Das könnte zum Beispiel sein, wenn eine Anfrage (HTTP-Request) nicht vollständig ist (z.B. fehlende Parameter) oder beim Löschen eines Fertigungsauftrags eine falsche Kennnummer übermittelt wird.

Als Basis für die Implementierung dient die über Moodle zur Verfügung gestellte Projektvorlage. Dabei handelt es sich um ein Maven-Multimodul-Projekt, welches aus einem Parent-, Client- und Server-Projekt besteht.

¹Level 2 nach dem [Richardson Maturity Model](#); siehe auch diese [StackOverflow-Frage](#)

Der Server soll mittels dem Framework Spring Boot entwickelt werden, es können aber auch andere Frameworks verwendet werden. Im Tutorial [Building a RESTful Webservice](#) wird ein einfacher RESTful Webservice entwickelt.

Zur Veranschaulichung muss mittels einem Werkzeug wie [Postman](#) jeweils ein valider und ein invalider HTTP Request pro Funktionalität abgesetzt werden und ein Screenshot davon gemacht werden.

Der **Client** simuliert eine Benutzerkonsole bei den Maschinen in der Werkhalle und muss folgende Ein- und Ausgaben über das Terminal ermöglichen:

- Ausgabe aller Fertigungsaufträge, sortiert nach ihrem Rang
- Fertigungsauftrag zur Bearbeitung an der Maschine auswählen

Die anderen Funktionalitäten des Servers werden nicht in der Benutzerkonsole benötigt. Der Client darf nur mit dem nativen [Java HTTP Client](#) auf den Server zugreifen. Daraus folgt die Abhängigkeit, dass mindestens ein JDK 11 verwendet werden muss.

Bewertung

Folgenden Mindestanforderungen muss diese Aufgabe genügen:

- Client und Server laufen getrennt und sind startbar
- Client und Server sind Maven-Projekte
- Alle Funktionen des Clients sind client- und serverseitig implementiert

Um die volle Punktezahl zu erreichen, müssen folgende Punkte umgesetzt sein:

- REST Semantik korrekt umgesetzt
- Alle Funktionalitäten sind implementiert
- Alle Überprüfungen werden durchgeführt und entsprechende Fehler mit richtigen HTTP-Statuscodes zurückgegeben

Dokumentation & Abgabe

Abzugeben ist eine einzelne Archivdatei (.zip) mit folgendem Inhalt:

- Exportiertes Projekt (.zip) oder exportierte Projekte (wenn Client und Server in separaten Projekten umgesetzt wurden). Es sind nur Textdateien erlaubt; binäre Dateien wie beispielsweise Java Bibliotheken, das `target`-Unterverzeichnis oder das kompilierte Projekt dürfen nicht in der Archivdatei enthalten sein.²
- Screenshots vom Zugriff auf die RESTful Webservices mittels einem Werkzeug

Deadline:

²Am besten `mvn clean` vor dem Erstellen des Zip-Archivs der Projekte ausführen.

- Deadline: Dienstag, 22. November 2022, 23:59 Uhr
- Format: Archivdatei im Format .zip
- Abgabe via [Moodle](#)

Startpunkt der Recherche

- Beschreibung des Szenarios
- TB3 - REST
- Übungsbeispiel
- <https://spring.io/guides/gs/rest-service/>
- <https://openjdk.java.net/groups/net/httpclient/intro.html>
- <https://docs.oracle.com/en/java/javase/11/docs/api/java.net.http/java/net/http/HttpClient.html>
- <http://zetcode.com/java/httpclient/>
- <https://dzone.com/articles/java-11-standardized-http-client-api>