

# Anwendungssysteme (SS 2017)

Prof. Dr. S. Tai, Dominik Ernst

## Bewertete Übungsaufgabe 3 – REST (10 Portfoliopunkte)

In dieser Aufgabe werden Sie eine Schnittstelle gemäß dem REST Architekturstil erstellen und die Nutzung von REST-basierten Services erlernen. Sie werden außerdem das Deployment von Anwendungskomponenten in virtualisierten Containern am Beispiel von Docker kennenlernen.

### Aufgabenstellung

#### a) REST-Schnittstellenbeschreibung und -Implementierung (6 Punkte)

Erstellen Sie eine REST-basierte Schnittstelle für den Empfang und das Auslesen von *Smart Meter Messdaten*. Nehmen Sie an, dass pro Smart Meter mehrere Messgrößen vorliegen und für jede Messgröße sekundengenaue Messwerte empfangen werden können. Umgekehrt soll die Schnittstelle auch das Auslesen der Messwerte sowohl für einen Smart Meter, als auch für einzelne Messgrößen, welche von einem Meter erfasst wurden ermöglichen. Die Ausgabe von Messwerten einer Messgröße soll jedoch nur als Durchschnittswert für ein 15-min Intervall erfolgen. Ihre Schnittstelle soll Daten im JSON-Format senden und empfangen. Ihre Schnittstelle soll außerdem alle Smart Meters und deren jeweiligen Metriken als Liste anbieten. Befolgen Sie für das Design der Schnittstelle die Vorgaben des REST-Architekturstils. Ihre Schnittstelle soll als Docker Container ohne besondere Konfiguration lauffähig sein.

#### b) REST-Client (4 Punkte)

Um Ihre Schnittstelle zu testen, steht Ihnen eine existierende Komponente namens *SMEmu* in Form eines Docker Containers zur Verfügung. Dieser verfügt über folgende Schnittstellenbeschreibung:

GET /meters

GET /meters/\*meterid\*

GET /meters/\*meterid\*/data

Das Image kann mit folgendem Befehl geladen werden:

```
docker pull gitlab-registry.tubit.tu-berlin.de/anwendungssysteme/rest-smemu
```

Erstellen Sie eine Client-Komponente, die Daten aus dem SMEmu Container liest und an Ihre REST-Schnittstelle sendet. Ist die Schnittstelle von SMEmu RESTful? Begründen Sie Ihre Antwort. Erfragen Sie mit Hilfe des Clients über Ihre API die durchschnittlich anliegende Stromstärke für ein beliebiges 15-min-Intervall für den Smart Meter mit der ID, die auf „2“ endet.

### **Schritt 1 – Teil a)**

Beschreiben Sie die Ressourcen, welche über Ihre REST-Schnittstelle verwaltet werden und wie Operationen auf Pfade abgebildet werden. Erläutern Sie den Aufbau der Schnittstelle und nennen Sie die jeweiligen HTTP-Methoden, welche Sie für einzelne Pfade anbieten wollen. Erstellen Sie zur Übersicht eine Matrix mit Pfaden und HTTP-Methoden wie im Beispiel auf Vorlesungsfolie 266 von Foliensatz 3. Befolgen Sie die Vorgaben des REST Architekturstils.

*Hinweis: Für die Abfrage von Durchschnittswerten können Sie von statischen, nicht-überlappenden 15-Minuten Zeitfenstern ausgehen, d.h. ein Durchschnittswert pro fixem 15min-Intervall bei 96 Intervallen pro Tag.*

### **Schritt 2 – Teil a)**

Implementieren Sie die Schnittstelle mit Hilfe von Spring Boot und erstellen Sie einen Docker-Container, in dem Sie Ihre Schnittstelle bereitstellen. Bauen Sie Ihre Anwendung auf das in ISIS zur Verfügung gestellte Basisprojekt auf.

*Hinweis: In der Übung bekommen Sie eine kurze Einführung in die Notwendigen Technologien. Alternativ greifen Sie auf eines der zahlreichen entsprechenden Online-Tutorials zurück, wozu Sie Links ebenfalls in den Übungsunterlagen finden. Wichtig: Ihre Implementierung muss das HATEOAS-Prinzip nicht unterstützen.*

### **Schritt 3 – Teil b)**

Erstellen Sie eine Client-Komponente, die Messdaten von einem SMEmu Container per http-Aufruf abfragen kann und diese an Ihren API-Container sendet. Ermitteln Sie experimentell das Verhalten des SMEmu-Containers, indem Sie mehrere Aufrufe an die beschriebenen Pfade durchführen. Übertragen Sie mindestens 1000 Messwerte.

*Hinweis: Nutzen Sie auch hier die Informationen aus der Übung um Anhaltspunkte für eine Vorgehensweise zu bekommen.*

### **Schritt 4**

Dokumentieren Sie die Ergebnisse Ihrer Arbeit in einer Präsentation (Powerpoint o.ä.). Diese Präsentation muss für das Gesamtverständnis Ihrer Lösung genügen, daher sollten relevante Codeauszüge enthalten sein. Das Deckblatt der Präsentation muss die Namen der beteiligten Gruppenmitglieder beinhalten.

## Schritt 5

Exportieren Sie die Präsentation als **PDF-Dokument(!)**. Packen Sie ihren kompletten Code aus Eclipse und alle zugehörige Artefakte als .zip-Datei. Für die Abgabe als \*.zip-Datei nutzen Sie wieder die „Exportieren“-Funktion von Eclipse (File -> Export -> Archive File).

Reichen Sie die Präsentation und die .zip-Datei mit Hilfe des ISIS-Systems ein.

Bewertungsgegenstand ist die Präsentation! Der Quellcode dient als Referenz und Nachweis der Leistung.

Wie immer genügt es, wenn die Einreichung durch ein Gruppenmitglied erfolgt.

**Deadline für den Upload: 16.07.2017, 23:55.**

Hinweise:

- Die Bearbeitung erfolgt in den gleichen Gruppen wie bisher.
- Verwenden Sie als Grundlage für ihr Projekt die in ISIS zur Verfügung gestellte Vorlage, die auch in der Übung kurz vorgestellt wird/wurde.
- Stellen Sie inhaltliche Fragen bitte im Forum des ISIS-Kurses, bei technischen Problemen (die Sie auch nach gründlicher Recherche nicht beheben konnten) wenden Sie sich an Ihren Tutor.
- Quellcode, der übernommen wird, ist zu kennzeichnen und die Quelle anzugeben.
- Nutzen Sie Javadoc-Kommentare für alle in Java implementierten Komponenten.