Implementierung von REST Services zur Umsetzung der SQLcoach Anwendung

Max Mustermann

Zusammenfassung
Das Abstract ist eine maximal 200 Worte lange Zusammenfassung des Inhalts der Arbeit, so dass sich der Leser vorab ein erstes Bild vom Inhalt machen kann.

Keywords
SQL-Coach, REST, Java, Service

Hochschule Kaiserslautern
Corresponding author: max.mustermann@hs-kl.de

Inhaltsverzeichnis

Anforderungen

Entfernen Sie bitte diesen Abschnitt aus Ihrem fertigen Dokument und arbeiten Sie den Inhalt in den Text mit ein. Ihr REST Service soll folgende User Stories abdecken.

1. Der REST-Service liefert auf Anfrage alle Szenarien mit allen Infos zu den Szenarien.
2. Der REST-Service liefert auf Anfrage alle Aufgabengruppen zu einem Szenario mit allen Infos zu den Gruppen.
3. Der REST-Service liefert auf Anfrage alle Aufgaben zu einer Aufgabengruppe mit allen Infos zu den Aufgaben.
4. Der REST-Service erlaubt das Hinzufügen von Szenarien, Aufgabengruppen und Aufgaben.
5. Der REST-Service erlaubt das Ändern der Eigenschafen zu Szenarien, Aufgabengruppen und Aufgaben.
6. Der REST-Service erlaubt das Löschen von Szenarien, Aufgabengruppen und Aufgaben.
7. Der REST-Service erlaubt das Anlegen, Ändern und Löschen von Trainingsdatensätzen mit frei konfigurierbaren Datasources.
8. Der REST-Service erlaubt das Auslesen der Tabelleninformation aller Tabellen in einem Trainingsdatensatz (Tabelleninformation = Tabellennamen, Spaltennamen, Primärschlüssel, Sekundärschlüssel).
9. Der REST-Service erlaubt das ausführung von beliebigen *SELECT* Abfragen auf den Trainingsdatensätzen.
10. Der Personaldatensatz von Prof. Schiefer muss mit eingebunden werden, so dass Ihr Service nach der Installation sofort einsatzbereit ist.

Weitere Anforderungen:

1. Voll dokumentierter Quellcode (jede *PUBLIC* Klasse und Methode hat sinnvollen JavaDoc)
2. Guter Programmierstil (insb. Namensgebung, Funktionen, Objekte, Exceptions, Logging usw.)
3. Zero Warnings bei SonarJava oder Intellij Code Inspection (oder Argumentation warum diese Warnung ignoriert werden kann)
4. Exakte Einhaltung der vorgegebenen REST Schnittstellen
5. Exakte Einhaltung der vorgegebenen maven/ docker Installation
6. Trainingsdatensätze in einer von den Stammdaten unabhängiger Postgres Instanz (innerhalb von Docker).

7. Nachvollziehbare Branches in Ihrem GIT (Namenskonvention: SCS-[NUM]-[feature/ bugfix /doc/ refact/ test]-name)

1. Einleitung

Die Einleitung sollte ausreichenden Hintergrund für den Leser liefern, so dass er sich ohne großes Studium von Sekundärliteratur in das Thema hineindenken kann.

Es sollen hier die grundlegenden fachlichen Begriffe eingeführt werden.

Auch sollte die Motivation für die vorliegende Arbeit dargelegt werden, sowie die Zielsetzung, die man erreichen will. Weiter wird auch das Thema von eventuell verwandten Themen abgegrenzt. Hier kann auch die Literatur [1, 2, 3] vorgestellt werden.

Als Beispiel können Sie hier ein Bild des SQLCoaches einbinden (Abbildung 1).

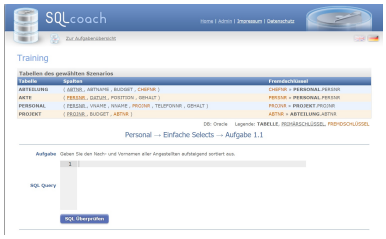

Abbildung 1. Die SQLcoach Anwendung (Quelle `https://sqlcoach.informatik.hs-kl.de/sqlcoach/`).

Folgende Punkte könnten hier angesprochen werden:

- Sinn und Zweck der Anwendung SQLCoach.
- Sinn und Zweck von REST Services.
- Einfüren der wichtigen Begriffe.
- Welche Problemstellung wird hier bearbeitet?

2. Methoden und Werkzeuge

Kurze Einführung in den Abschnitt. Es wird beschrieben was jetzt kommt, d.h. welche Methoden und Werkzeuge im folgenden vorgestellt werden. Es wird noch nicht darauf eingegangen was im Ganzen damit gebaut wurde. Es werden nur die Details zu den einzelnen Werkzeugen angegeben, so dass im nächsten Kapitel darauf Bezug genommen werden kann.

2.1 JAX-RS

JAX-RS wurde im Zuge der Tutorialreihe ”REST Web Servicesäls Java Programmierschnittstelle kennengelernt. Verschiedene Annotationen von JAX-RS wurden verwendet. Pfad Annotationen, welche die Url der Schnittstelle definieren, der Requesttyp, welcher die Art der Anfrage definiert, sowie @Consumes und @Poduces, welche die MIME-Typ der jeweiligen akzeptierten bzw. zurückgegebenen Ressource definieren. Wie wurde von dieser Technologie verwendet? Welche technischen Artefakte (Dateinamen, Konfiguration, Klassen,...) sind entstanden, wie heißen diese und was machen diese.

2.2 Docker

Zwei Sätze zur Einführung der Technologie. Was wurde von dieser Technologie verwendet? Wie wurde von dieser Technologie verwendet? Welche technischen Artefakte (Dateinamen, Konfiguration, Klassen,...) sind entstanden, wie heißen diese und was machen diese.

2.3 Maven

Zwei Sätze zur Einführung der Technologie. Was wurde von dieser Technologie verwendet? Wie wurde von dieser Technologie verwendet? Welche technischen Artefakte (Dateinamen, Konfiguration, Klassen,...) sind entstanden, wie heißen diese und was machen diese.

2.4 Postgres

Zwei Sätze zur Einführung der Technologie. Was wurde von dieser Technologie verwendet? Wie wurde von dieser Technologie verwendet? Welche technischen Artefakte (Dateinamen, Konfiguration, Klassen,...) sind entstanden, wie heißen diese und was machen diese.

2.5 Apache Log4j

Zwei Sätze zur Einführung der Technologie. Was wurde von dieser Technologie verwendet? Wie wurde von dieser Technologie verwendet? Welche technischen Artefakte (Dateinamen, Konfiguration, Klassen,...) sind entstanden, wie heißen diese und was machen diese.

2.6 GIT

Zwei Sätze zur Einführung der Technologie. Was wurde von dieser Technologie verwendet? Wie wurde von dieser Technologie verwendet? Welche technischen Artefakte (Dateinamen, Konfiguration, Klassen,...) sind entstanden, wie heißen diese und was machen diese.

3. Ergebnisse

In diesem Abschnitt beschreiben Sie nun das Endergebnis Ihrer Arbeit.

3.1 Architektur

Die beziehen sich hier auf die im Methodenteil vorgestellten Artefakte und beschreiben wie diese zusammengesetzt wurden. Welche Komponenten ruft welche Komponente auf?

Sie müssen hier auf jede der Methoden in Abschnitt 2 Bezug nehmen.

3.2 REST Schnittellen

Bitte beschreiben Sie hier auch die von Ihrer Komponenten bereitgestellten REST Schnittstellen, da diese das Endergebnis Ihrer Arbeit darstellt.

3.3 Installation

Geben Sie hier eine Installationsanleitung. Ihre Software muss Sie wie folgt bauen und starten lassen:

```
git clone MYREPO

mvn clean package

docker-compose  build

docker-compose  up
```

4. Diskussion

In diesem Abschnitt werfen Sie einen kritischen Blick auf Ihre Arbeit:

1. Haben Sie alle angeforderten User Stories umgesetzt?
2. Welche Features haben sie zusätzlich umgesetzt?
3. Welche positiven Eigenschaften hat Ihre Implementierung?
4. Welche negativen Eigenschaften hat Ihre Implementierung?
5. Sind die Schnittstellen gut gelungen? Warum Sind das gute Schnittstellen?
6. Wie performant ist Ihre Implementierung?
7. Welche Best Practices haben Sie eingehalten?
8. Wie hoch ist Ihre Test Coverage (JUnit)?
9. Wie ist die Code-Qualität? Berichten Sie hier über das Ergebnis der Code-Inspection (SonarJava oder IntelliJ IDEA Code-Inspection).

Dieser Abschnitt ist nicht leicht zu schreiben, aber dieser Abschnitt ist essentiell für Ihre Benotung. **Sie müssen hier das Positive herauszustellen!** Zudem sollen Sie hier auch auf Unzulänglichkeiten Ihrer Arbeit eingehen. Wenn Sie die Schwachstellen Ihrer Arbeit hier beläuchten und idealerweise Argumente finden warum das so gelöst wurde, dann hat einen sehr positiven Einfluss auf die Benotung.

5. Zusammenfassung

Hier wird nochmal der Inhalt und die Ergebnisse der Arbeit kurz zusammen gefasst. Des Weiteren werden Themen und Aufgabenstellungen genannt, die es lohnt weiter zu untersuchen.

Literatur

[1] David Harel. Statecharts: A visual formalism for complex systems. *Sci. Comput. Program.*, 8(3):231–274, June 1987.

[2] D. Harel, Y. Feldman, and M. Krieger. *Algorithmik*. Springer Berlin Heidelberg, 2006.

[3] Jilles Van Gurp and Jan Bosch. On the implementation of finite state machines. In *in Proceedings of the 3rd Annual IASTED International Conference Software Engineering and Applications, IASTED/Acta*, pages 172–178. Press, 1999.

Erklärung zur Ausarbeitung

Hiermit erkläre ich, Daniel Braun(880335), dass ich die vorliegende Ausarbeitung selbstständig und ohne fremde Hilfe angefertigt habe und keine anderen als in der Abhandlung angegebenen Hilfen benutzt habe; dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe. Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben kann.

_____

Unterschrift