

Verteilte Softwaresysteme

Blatt 4

Prof. Dr. Oliver Braun

Fakultät für Informatik und Mathematik
Hochschule München

Letzte Änderung: 28.11.2018 16:20

Abgabe bis 15.01.2019, 11:45 per Pull-Request gegen den master-Branch.

Aufgabe — Multiplex-Kino

Die fristgerechte Abgabe einer Lösung für diese Aufgabe ist notwendige Voraussetzung für den Leistungsnachweis.

Diese Aufgabe können Sie wieder alleine oder zu zweit bearbeiten und abgeben. Über den GitHub-Classroom-Link <https://classroom.github.com/g/GunEmM3r> bekommen Sie das Repository und die Teams. Nachdem Sie auch in anderer Zusammensetzung als auf Blatt 3 arbeiten können, müssen Sie für Blatt 4 wieder ein neues Team erstellen.

Wenn Sie alleine arbeiten, dann gehen Sie jetzt nach dem Git Flow oder nach dem GitHub-Flow vor, zu zweit nach dem Git-Flow.

Entwerfen Sie eine Verwaltung für ein Multiplex-Kino als verteilte Anwendung und Implementieren Sie einen Prototypen als *Proof of Concept*. Mit dem Proof of Concept sollen Sie im Wesentlichen zeigen, dass das Zusammenspiel der Services funktioniert. Es ist nicht notwendig als *Fleißarbeit* alles zu implementieren. Im Zweifelsfall fragen Sie mich. Zeigen Sie aber mindestens das Anlegen einer Vorstellung und die Reservierung von Kinokarten für eine bestimmte Vorstellung und bestimmte Plätze; beides inklusive der notwendigen Fehlerbehandlung, z.B. Plätze schon besetzt oder Film gibt es gar nicht.

Vorgaben

Der Zugriff auf die Anwendung erfolgt per REST- oder GraphQL-API auf einen zentralen Service.

Die Verwaltung selbst besteht mindestens aus weiteren Services für die

- Kinosaalverwaltung
- Benutzerverwaltung
- Filmverwaltung
- Vorstellungsverwaltung
- Reservierungsverwaltung
- Protokollierung der Zugriffe durch einen Loggerservice

Die Services werden durch Remote-Actors unter Verwendung von [Proto.actor](#) implementiert. Jeder Service ist ein eigenes ausführbares Programm. Innerhalb der Services wird auch alles nach dem Actor-Modell implementiert. Die Kommunikation der Services miteinander erfolgt per [Protocol Buffers](#).

Alle Services protokollieren Ihre Schritte durch Nachrichten an den Logger, der diese Informationen in einer Datei speichert.

Bestandteile der verwalteten Daten

- Kinosäle bestehen mindestens aus einem Namen, z.B. "Kino 1", und einer Repräsentation der vorhandenen Sitzplätze.
- Benutzer haben mindestens einen Namen und eine Verknüpfung zu ihren Reservierungen.
- Filme haben mindestens einen Titel, eine Länge, einen Wert, wie oft der Film gezeigt werden darf, sowie einen Zähler wie oft der Film gezeigt wird. Beim Anlegen einer Veranstaltung muss sichergestellt werden, dass der Film überhaupt noch ein weiteres Mal gezeigt werden darf.
- Eine Vorstellung verknüpft einen Film mit einem Kinosaal. Stellen Sie bereits beim Anlegen einer Vorstellung sicher, dass in einem Kinosaal keine überlappenden Vorstellungen erzeugt werden. Rechnen Sie dabei Zeit für Werbung sowie Zeit für die Besucher zum Kommen und Gehen ein.
- Eine Reservierung verknüpft einen Benutzer mit einer Vorstellung. Eine Reservierung kann gleich mehrere Sitzplätze enthalten. Eine Reservierung kann vor Beginn des Filmes kostenfrei storniert werden, aber nur, wenn sie noch nicht bezahlt ist. Ist eine Reservierung 30 Minuten vor Beginn eines Filmes noch nicht bezahlt, verfällt sie und die reservierten Sitzplätze können anderweitig vergeben werden. Sie müssen das nicht aktiv machen, sondern nur dann, wenn im entsprechenden Zeitfenster eine Anfrage nach freien Plätzen gemacht wird.

Abzugebende Bestandteile

- Go-Code as usual (dokumentiert und getestet).
- Anbindung an den Travis CI.
 - Ausführen der Tests auf dem Travis CI.
 - Generierung der Godoc auf dem Travis CI und Veröffentlichung als GitHub-Page.
- Beschreibung der Kommunikation zwischen den Services in der Datei `Protocol.md`.
- Ablaufbeschreibung von mindestens 3 Use Cases, bei denen mehrere Services beteiligt sind, in der Datei `UseCases.md` unter Verwendung von UML Interaktionsdiagrammen, die Sie als [Images](#) einbinden.

Abnahme

Im Rahmen des Praktikums erfolgt die Abnahme durch eine kurze Vorführung und Erklärung Ihrer Bibliothek auf dem Beamer.