

Buchungs-App für einen Reiseanbieter

Entwicklung eines Backends mit skalierbarer Microservice-Architektur

Studenten

Christoph Scheiwiller

Einleitung: Das Bedürfnis nach Erlebnissen in der Natur ist für viele eine Möglichkeit die persönliche Freiheit zu entfalten. Viele Vanlifer würden gerne in ihren Camper einsteigen, unbeschwert drauflosfahren und spontan irgendwo Übernachten. Da Wildcampen vielfach verboten ist und sich die Suche nach einem Stellplatz oft schwierig gestaltet, tritt häufig bereits vor dem Reisestart der erste Frust auf. Die Vanlife Travel GmbH hat sich zum Ziel gesetzt, das Planen eines Campingausflugs einfacher zu machen. Dafür soll eine Webapplikation erstellt werden, welche Anbieter und Kunden zusammenbringt und den Kunden während der ganzen Reise begleitet.

Michael Hofmann

Aufgrund des Umfangs der Studienarbeit arbeiten zwei Teams daran, die Arbeit wurde in Frontend und Backend aufgeteilt. Diese Arbeit umfasst den gesamten Backend-Teil des Systems.

Ergebnis: Das Backend sollte für viele Benutzeranfragen ausgelegt sein, da der Kunde plant, den gesamten europäischen Markt zu bedienen. Deshalb fiel der Architekturentscheid im Backend zugunsten einer Microservice-Architektur aus, mit dem Ziel die einzelnen Services möglichst unabhängig voneinander zu gestalten und nach Bedarf mehrere Instanzen starten zu können. Die Microservices werden dazu nach Geschäftsbereichen aufgeteilt und durch unterstützende Services ergänzt, dadurch kann die Entwicklung auch auf mehrere Entwicklerteams aufgeteilt werden, welche nur geringe Abhängigkeiten voneinander haben.

Für das Frontend soll dabei nicht ersichtlich sein, welche Microservices im Backend aufgerufen werden. Weiter soll durch die Architektur keine merkbare Verzögerung zwischen Request und Response auftreten. Um diesen Anforderungen gerecht zu werden, wurde auf das gRPC Framework gesetzt, da dessen Nachrichten schlank sind und effizient übertragen werden. gRPC hat zudem den Vorteil, dass es technologisch unabhängig ist.

Alle Microservices wurden in Java unter Verwendung des Spring Boot Frameworks implementiert. Für die Automatisierung des Build-Prozesses wird Gradle mit Kotlin DSL eingesetzt. Als Datenbank wurde PostgreSQL und für die Persistenz JPA verwendet. Die Datenbanken werden dabei mithilfe von Docker Compose automatisiert erstellt und beim Start der Applikation wird automatisch eine Datenbankmigration mit Flyway durchgeführt, das garantiert die Kompatibilität zwischen Applikationen und Datenbank. Für das Login wurde auf OAuth 2.0 gesetzt, um dem User grösstmögliche Sicherheit und einen einfachen Login-Prozess zu bieten. Als Standard für die Beschreibung des HTTP API wird OpenAPI verwendet.

Referent

Prof. Dr. Daniel Patrick Politze

Themengebiet
Software

Projektpartner

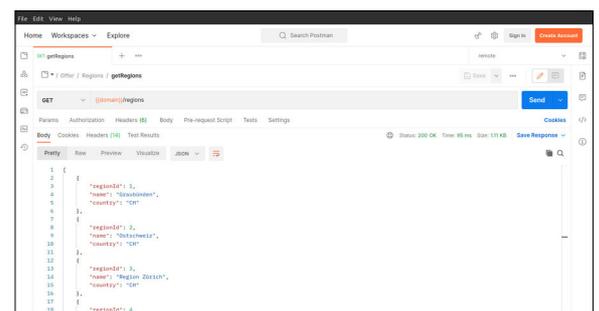
Vanlife Travel GmbH,
Zürich, ZH

Fazit: Aufgrund der umfangreichen Anforderungen wurden die wesentlichen Anforderungen identifiziert und implementiert, wodurch die technische Machbarkeit nachgewiesen werden konnte. Dank der Microservice-Architektur ist das System gut skalierbar und die Modularisierung ermöglicht es neue Anforderungen mit geringem Aufwand zu implementieren.

Eingesetzte Technologien Eigene Darstellung



Anfordern einer Ressource vom Server mit Postman Eigene Darstellung



Datenmodell des Systems Eigene Darstellung

