

Implementation

TravelCompats

Technology
Arts Sciences
TH Köln

Installationsanleitung TravelCompats

Server

1. **NodeJS** downloaden und installieren.
 - <https://nodejs.org/en/download/>
2. **MongoDB** downloaden und installieren.
 - <https://www.mongodb.com/download-center>
- 3.

Client

Abweichungen vom Konzept

Von Open Street Maps zu Google Places

Ressourcen die zur Ermittlung der Atmosphäre von Orten gesammelt und verarbeitet werden müssen, sollen laut Konzept aus der Bibliothek von Open Street Maps stammen. Zu Beginn war geplant alle Informationen von OSM über REST Methoden abzufragen. Nach einigen Evaluationsschritten während der praktischen Programmierarbeit wurde allerdings leider ersichtlich, dass dies scheinbar nur über schwierige Umwege möglich ist. So konnte man in den Testphasen mit dem OSM-Webservice keine Informationen als JSON-Objekt beziehen. Dies könnte nur umgangen werden, wenn ein OSM Dienst auf dem Client im Hintergrund ausgeführt wird, Informationen an den Server schickt, wo dieser verarbeitet werden und im Anschluss zurück an den Client fließen. Diese Alternative ist allerdings sehr unperformant und wird aus dem Entwicklungsprozess ausgeschlossen.

Als Konsequenz werden nun Daten von der Google Places API bezogen. Diese bietet zwar nicht alle benötigten Informationen und manche sind erst nach Zahlung mit dem Premium Modell erhältlich, dennoch eignet sich dieses Verfahren um mit der Implementation fortzufahren.

Aktuelle Attribute der Atmosphäre

Durch die Profilangaben eines Nutzers kann eine gewünschte Atmosphäre ermittelt werden. Ein Ort verfügt durch seine Attribute bereits über jene. So prüft die Anwendungslogik also ob sich die jeweiligen Atmosphären abgleichen. Passt eine Ort-Atmosphäre zu einer gewünschten Nutzer-Atmosphäre wird dieser Ort mit hoher Wahrscheinlichkeit empfohlen.

Aktuelle Attribute sind:

- **Geruch** eines Ortes.
- **Geschmack** des Essens z.B.
- **Freundlichkeit** der Menschen und des Personals

- **Geeignete Aktivitäten.** Stammt aus Analyse von App Daten
- **Geräusche** (Umgebungs lautstärke) stammen aus Netatmos und eigenen Aufnahmen, um zu prüfen wann Richtwerte überstiegen werden.
- **Gesprochene Sprachen** an einem Ort und eines Landes werden berücksichtigt.
- **Wetter.** Informationen stammen von Open Weather Map.
- **Familientauglichkeit**
- **Sauberkeit** von Straßen und Gebäuden.
- **Behindertenfreundlichkeit/ Barrierefreiheit** (Differenzierbar in verschiedenen Bereichen wie Farbenblindheit und Rollstuhltauglichkeit)
- **Sehenswürdigkeiten.**
- **Textuelle Bewertungen** von anderen Nutzern über gelistete Orte in Google Places werden bei Empfehlungen mit angezeigt. Ein Keyword-Filter wird hierbei eingesetzt, um aus einem Fließtext wertvolle Informationen zu sammeln.

Diese Attribute werden in dezimalen Werten beschrieben - von 0 bis 5. 0 ist dabei ein Platzhalter, falls dieses Attribut gar nicht vertreten ist. 5 steht für den stärksten positiven Einfluss.

Daten aus installierten Nutzer-Apps sammeln

Wie Im Konzept beschrieben sollen Informationen von installierten Apps eines Nutzers dazu beitragen die Empfehlungs-Algorithmen zu verfeinern.

Um dies zur Code-Implementierung zu simulieren, werden nun keine Daten aus MyFitnessPal bezogen, sonder aus der Google Fit App. Grund dafür ist der fehlende API-Schlüssel von Seiten MyFitnessPals. Dieser wurde zwar beantragt, doch noch nicht ausgeliefert und wird offensichtlich nicht rechtzeitig eintreffen.

Android Versionskontrolle

Im Code ist zu erkennen, dass sich die Versionsabfrage auf Android 6 bezieht. Trotzdem ist die Software und die spezifische Methode auch auf älteren Systemen, z.B. 5.1. funktionstüchtig.

Der Grund für die Abfrage der neuesten Versionsnummer ist lediglich notwendig, weil die Methode und Syntax der aktuellen Android Studio Version danach verlangt, wenn man die Ortungsfunktion nutzen möchte.

CompyMarks

Weil sich der Workload auf die Anwendungslogik der Empfehlungssysteme und Atmosphären-Ermittlung konzentriert hat