

Geolocation-basierte Gamification der BERNMOBIL-Leitdaten



BERNMOBIL stellen ihre Echtzeit-Leitdaten über eine HTTP-Schnittstelle im Protocol Buffer-Format zur Verfügung.

Wir wollen diese Daten, davon vor allem die Positionsinformationen von Fahrzeugen, deren Typ und spezielle Merkmale, zusammen mit der Geoposition des Users verwenden, um eine Challenge-basierte Gamification von Einzelfahrten zu ermöglichen.

AUSGANGSLAGE

Die begrenzte Zeit lässt es nicht zu, das komplette Featureset zu implementieren. Darum reduzieren wir uns auf ein MVP mit folgenden Features:

- Mobile Web Frontend mit Kartenansicht, welche alle Haltestellen in der Umgebung anzeigt. Für die Zukunft vorbereitet wäre auch die Möglichkeit, Echtzeitdaten zu den Fahrzeugen anzuzeigen.
- Challenge-Ansicht, die das ansehen, starten und abschliessen von vordefinierten Challenges ermöglicht.
- Backend, welches die Leitdaten von der BERNMOBIL API abholt und für das Frontend aufbereitet.

TECHNISCHE IMPLEMENTATION

Das Frontend wird auf *Vue.js*, *Vuex* und *Bootstrap 4* gebaut und implementiert *leaflet* als Framework für die Anzeige von OpenStreetMap Karten.

Das Backend basiert auf *Node.js* mit *express* und einer *mysql*-Datenbank. Der Datenbankzugriff wird durch die Libraries *Knex.js* und *Bookshelf.js* gewährleistet. Um die Daten der BERNMOBIL-API zu konsumieren verwenden wir *protobuf.js*.

Für vereinfachtes Deployment wird die Datenbank in einem Docker-Image verpackt.

Endpoints

/challenges – Liste alle Challenges mit Regeln

/vehicles – Liste alle BERNMOBIL-Fahrzeuge und deren Standort

/stops – Liste alle BERNMOBIL-Haltestellen und deren Standort

Die Endpoints **/vehicles** und **/stops** akzeptieren als GET-Parameter jeweils **lat**, **lon** und **radius**, um die Anzahl Objekte basierend auf dem Umkreis des Standorts einzugrenzen.

Nennenswertes

Für unsere Implementation besonders nennenswert halten wir die umfangreiche Umsetzung von Geolocation-Diensten, insbesondere Standort-basierte Filter. Somit werden jeweils nur die Daten abgefragt und gezeichnet, die jeweils für den Standort relevant sind. Ebenfalls erwähnenswert ist der Einsatz von einem modernen Backend-Stack mit *Express.js* und Docker und die Verwendung von reaktiven Frontend-Technologien.

OFFENE PUNKTE / ABGRENZUNG

Bewusst weggelassen haben wir die Erkennung, ob sich ein Mobilgerät in einem fahrenden Verkehrsmittel befindet. Wir halten dies für ein nicht sehr komplexes, jedoch aufwändiges Feature, welches wir schlecht auf der BärnHäckt-Bühne präsentieren können.

REPOS

Frontend: <https://github.com/Bufalom/bernhackt2018-aldente>

Backend: <https://github.com/Andy-Bigler/bernhackt2018-aldente-backend>