



Produktbeschreibung Backend



Produktbeschreibung - Taxi-Tool

Autor: Daniel Altenburg

Version: 1.0

Veröffentlicht am: 25.01.2024

Produktinformationen:

Backend:

- Java: open JDK 17
- Spring Boot: 3.2.0
- Spring Security: 6.2.0
- Apache Maven: 4.0.2
- Datenbank: Relational (H2 Nur für Demoversion)
- Testumgebung: Unit / Mockito
- Graphopper:7.0

Frontend:

- Angular: 16.2.7
- TypeScript: 5.1.6
- Node: v18.18.1
- Npm:10.2.1

Zielgruppe:

In Deutschland, gerade im ländlichen Bereich, spielen Krankentransporte zu Kliniken, sowie Pauschalfahrten eine entscheidende Rolle in der kommerziellen Personenbeförderung. Für diese Fahrten außerhalb des normalen Taxibetriebs können auch Fahrzeuge ohne Taxameter und Taxischild eingesetzt werden, da direkt über die Versicherer abgerechnet wird. Dies senkt die Anschaffungs- und Betriebskosten des Fahrzeugs und ist vor allem für junge Unternehmen interessant.

Dieses Programm wurde speziell für Unternehmen entwickelt, die vor allem im sogenannten Mietwagenverkehr tätig sind.

Details:

Es ist wichtig zu betonen, dass Krankentransporte oft im Zusammenhang mit onkologischen Behandlungen stehen. Diese onkologischen Behandlungsfahrten haben in der Regel zahlreiche Anschlussfahrten, die sorgfältig koordiniert werden müssen. Die Anschlusstermine werden vom Fahrer oder direkt vom Fahrgast an die Zentrale übermittelt. Die Weitergabe der Termine an die Zentrale muss lückenlos und zeitnah erfolgen.

Das Programm Taxi-Controller hilft dabei Fehler bei der Übermittlungskette zu vermeiden und nimmt diese zeitnah entgegen.

Aufgaben des Taxi-Tool:

- Das erzeugen und übergeben neuer Fahrten durch den Fahrer oder Kunden.
- Zeiterfassung der Fahrer
- Start und Zieldaten auswerten zur Fahrzeitberechnung.

Technische Details:

Der Taxi-Tool ist eine geteilte webbasierte Fullstack-Anwendung, die speziell für kleine Taxiunternehmen und Fahrdienstleister in Deutschland entwickelt wurde. Sie besteht aus einer Serverseitigen Backend-Applikation sowie eine Applikation für Mobile Geräte die über ein JSON Datenformat kommunizieren.

Erweiterungs- Kombinationsmöglichkeiten:

Die Struktur ist so aufgebaut, die Backendanwendung in wenigen Schritten durch Klassenübernahme in die Anwendung Taxi-Controller integriert werden kann.

Fahrzeitberechnung über ein KI Modell das auf Maschine-Learning und ein Expert-System aufgebaut wird.

Httpie Befehle

Admin Bereich (Achtung zum Testen muss der Datensatz „Freiburg erst angelegt werden)

```
http PUT localhost:8080/updateTripInfo/Freiburg \  
  gebiet=Berlin \  
  grundgebuer=NeueGrundgebuer \  
  kilometerpreis=NeuerKilometerpreis \  
  anfahrt=NeueAnfahrt \  
  grossraum=NeuerGrossraum --auth admin:adminPass
```

-

```
http GET localhost:8080/getAllTripInfo --auth admin:adminPass
```


```
http GET localhost:8080/getTripInfo/Freiburg --auth admin:adminPass
```


```
http POST localhost:8080/createTripInfo \  
  gebiet=Freiburg \  
  grundgebuer=5.00 \  
  kilometerpreis=3.50\  
  anfahrt=10.0 \  
  grossraum=10.0 \  
  --auth admin:adminPass
```

Trip Service Bereich

```
http --auth admin:adminPass POST localhost:8080/calculateTrip \  
  startLat:=47.69224288396932 \  
  startLon:=8.256928640765006 \  
  endLat:=48.005639081823865 \  
  endLon:=7.885851816065002 \  
  requestGebiet=Waldshut-Tiengen \  
  requestAnfahrt=N \  
  requestGrossraum=J
```

```
http --auth admin:adminPass POST \  
localhost:8080/calculateDepartureTime \  
  startLat:=47.69224288396932 \  
  startLon:=8.256928640765006 \  
  endLat:=48.005639081823865 \  
  endLon:=7.885851816065002 \  
  requestTermin=12:01 \  
  requestLaufweg=10 \  
  requestPuffer=20
```

Freiburg ITZ 48.005639081823865, 7.885851816065002
Maps: Entfernung 68,9 Zeit ca.66 min

NEW Trip Bereich

http GET localhost:8080/getNewTrip --auth admin:adminPass

http POST http://localhost:8080/setNewTrip \
 abholort="Ringstrasse 14 78976 Friedheim" \
 anmerkung="keine" \
 fahrer="0000" \
 fahrgast="Siegfried Meyer" \
 gebucht="NEIN" \
 zeit="30.04.24, 08:00" \
 zielort="Freiburg Urologie" --auth admin:adminPass

Mitarbeiter Bereich

http POST localhost:8080/setTaxiStamp personalnummer=1212 --auth
admin:adminPass

http GET localhost:8080/searchTaxiStamp --auth admin:adminPass

Kunden Bereich

http GET
"localhost:8080/searchCustomerTrips?name=Daniel%20Altenburg" --auth
admin:adminPass

Kunden Login

http GET localhost:8080/getAllCustomerLogs --auth admin:adminPass

http PUT http://localhost:8080/updateCustomerLogs \
 name="Daniel Altenburg" \
 username="customer" \
 password="Hallo" \
 roles="CUSTOMER" \
 --auth admin:adminPass

http POST http://localhost:8080/createCustomerLog \
 name="Daniel Altenburg" \
 username="neuerBenutzername" \
 password="neuesPasswort" \
 roles="neueRollen" \
 --auth admin:adminPass

Mitarbeiter Login

http GET localhost:8080/getAllEmployeeLogs --auth
admin:adminPass

http PUT http://localhost:8080/updateEmployeeLogs \
 personalID="1212" \
 username="neuer Name" \
 password="neues Passwort" \
 roles="ADMIN" \
 --auth admin:adminPass

http POST http://localhost:8080/createEmployeeLog \
 personalID="1212" \
 username="neuerBenutzername" \
 password="neuesPasswort" \
 roles="neueRollen" \
 --auth admin:adminPass