

Hund ist nicht Katze

Fachmodul ÖV Journey Planning

Christian Bühler Flavio Tobler

August 4, 2018

1 Abstract

Abstract halt

Inhaltsverzeichnis

1	Abstract	I
2	Einleitung	2
3	Augabenstellung	3
4	Voraussetzungen	4
4.1	OpenTripPlanner	4
4.2	ConnectionScanAlgorithmus	4
4.3	General Transit Feed Specification	4
5	Endprogramm	5
5.1	Datenstruktur	5
5.1.1	TimeTable	5
5.1.2	StopCSA	5
5.1.3	TripCSA	5
5.1.4	FootpathCSA	5
5.1.5	ConnectionCSA	5
5.1.6	LegCSA	5
5.1.7	JourneyPointer	5
5.1.8	Journey	5
5.2	Programmablauf	5
5.3	TimeTableBuilder	5
5.4	Server	5
5.5	Webseitenaufruf	5
5.6	CSA	5
5.7	JourneyToTripPlanConverter	6
6	Programmierverlauf	7
6.1	OTP mit Schweizer Daten implementieren	7
6.2	Kann OTP ohne .osm file ausgeführt werden?	7
6.3	Modellierung der Datenstruktur	7
6.4	Automatische generierte Klassendiagramme	7
6.5	Dummy-GTFS Daten erstellen	7
6.6	Mocking	7
6.6.1	CSAMock	7
6.6.2	TimeTableBuilderMock	7
6.6.3	JourneyToTripPlanConverterMock	7
6.6.4	Probleme	7
6.7	TimeTableBuilder	7
6.8	JourneyToTripPlanConverter	7
6.8.1	Knackpunkte	8

7	Ergebniss	9
8	Fazit	10

2 Einleitung

Einleitung: Nutzen und Sinn des Projektes + Grobe Kapitelübersichtsbeschreibung der Arbeit

3 Aufgabenstellung

Unsere BA Aufgabenstellung

4 Voraussetzungen

Um unsere Programm und unser vorgehen zu besser verstehen muessen wir zuerst das von uns verwendete Basisprogramm sowie den von uns verwendeten Basisalgorithms erlaeutern.

4.1 OpenTripPlanner

Erklärung des OTP: Funktion, Entwicklungsgruppe, Code-Grob, Eingabeparameter

4.2 ConnectionScanAlgorithmus

Erklärung des Konzeptes und des CSA sowie dessen Optimierungen.

4.3 General Transit Feed Specification

Erklärung was GTFS ist und was dessen aufbau und regeln sind

5 Endprogramm

Dies ist eine Erläuterung des Endprogramms

5.1 Datenstruktur

Datenstrukturen des TimeTables und des Journeys erläutern. Sagen dass CSA hinter namen. Auch sagen was und wieso diese containerklasse

5.1.1 TimeTable

5.1.2 StopCSA

5.1.3 TripCSA

5.1.4 FootpathCSA

5.1.5 ConnectionCSA

5.1.6 LegCSA

5.1.7 JourneyPointer

5.1.8 Journey

5.2 Programmablauf

TimetableBuilder-;Serverstart-;Webseitenaufruf-;CSA-;JourneyToTripPlanConverter

5.3 TimeTableBuilder

Erstellte einen Timetable aus GTFS-Daten

5.4 Server

Startet server welchen die Webseite für aufrufe zur verfügung stellt

5.5 Webseitenaufruf

Aufruf über webseite abgesetzt. PlannerResource primärer einstiegspunkt

5.6 CSA

Eigentlicher algorithmus welcher journeys bildet

5.7 JourneyToTripPlanConverter

Wandelt vom algorithmus zurückgegebene Journeys in vom Server verlangten TripPlan um

6 Programmierverlauf

6.1 OTP mit Schweizer Daten implementieren

Routetype 1700 -j Miscellaneous Service + PC-Auslastung

6.2 Kann OTP ohne .osm file ausgeführt werden?

6.3 Modellierung der Datenstruktur

Java container klassen + UML Diagramm

6.4 Automatische generierte Klassendiagramme

6.5 Dummy-GTFS Daten erstellen

6.6 Mocking

Weshalb mocking?

6.6.1 CSAMock

einlesen von Timetable + Rückgabe eines Journeys

6.6.2 TimeTableBuilderMock

manuelles erstellen eines Timetableobjektes

6.6.3 JourneyToTripPlanConverterMock

auslesen und speichern eines JSON der Rückgabe einer Response der original Software + manuelles nachbauen erst ohne laufwege und umsteigen dann immer mehr dazu AgencyAndId + From/to erstellt aber nicht ins leg hinzugefügt + walksteps erstellen + startzeit stopzeit erstellen Gregorian Calendar

6.7 TimeTableBuilder

einstiegspunkt in GTFSMODUL + Namenskonflikt mit der onebusaway Bibliothek + Problematik Objektreferenz

6.8 JourneyToTripPlanConverter

bildet TripPlan aus Journey + Schleife für jedes Journey Itinerary

6.8.1 Knackpunkte

Walkdistance aus Koordinaten berechnen (lon,lat) + Himmelsrichtung aus Koordinaten berechnen
Timezoneoffset für Footpath + Start-, Stoptime für Footpath + LegGeometry aus Koordinaten berechnen
Walksteps erstellen + Datetime aus Time von Timetable, Date from request zusammensetzen

7 Ergebniss

Unsere Tests und deren Ergebnisse darstellen

8 Fazit

Fazit der BA über den CSA und das Projekt selbst

Abbildungsverzeichnis