

Graphserver, Graphserver Tools & Transitfeed

Installation

Um Graphserver, Graphserver Tools sowie die Transitfeed Tools nutzen zu können müssen diese installiert werden. Grundsätzliche sollte die Installation auf jedem Unix-artigen Betriebssystem möglich sein.

1. *graphserver_v*.7z* entpacken, in den Ordner *graphserver_v*/pygs* wechseln und mit dem Aufruf
`$ sudo python setup.py install`
Graphserver installieren.
2. *graphserver_tools_v*.7z* entpacken, in den Ordner *graphserver_tools_v** wechseln und ebenfalls mit
`$ sudo python setup.py install`
Graphserver Tools installieren.
3. Die aktuelle Version von transitfeed (<http://code.google.com/p/googletransitdatafeed/>) herunterladen, entpacken und im entsprechenden Ordner mit
`$ sudo python setup.py install`
installieren.

Die Archive und entpackten Ordner können anschließend einfach gelöscht werden.

Nutzung

Folgende Funktionalitäten werden bereitgestellt:

- Erstellen eines GTFS-Feeds aus einer Netzdatei (Visum)
- Überprüfung von GTFS-Feeds auf (grobe) Fehler
- Ansicht eines GTFS-Feeds als Layer über Google Maps
- Das Berechnen von kürzesten Wegen innerhalb eines GTFS-Feeds bzw. Netzdatei mit Hilfe von OpenStreetMap Daten

Netzdatei nach GTFS-Feed

Zur Umwandlung wird das Skript *gst_netToGtfs* benötigt. Es kann wie folgt aufgerufen werden:

```
$ gst_netToGtfs <input> <output>
```

Input ist die umwandelte Netzdatei und *output* der Name das zu erstellenden GTFS-Feeds. Koordinaten innerhalb der Netzdatei müssen immer in WGS84 vorliegen.

Überprüfen eines GTFS-Feeds

Hierzu steht das Skript *feedvalidator.py* zur Verfügung. Es wird wie folgt aufgerufen:

```
$ feedvalidator.py <feed>
```

Feed steht natürlich für der zu überprüfenden GTFS-Feed. Nach durchlaufen der Tests, wird eine Webseite erstellt und angezeigt, die das Ergebnis enthält.

Ansicht eines GTFS-Feeds

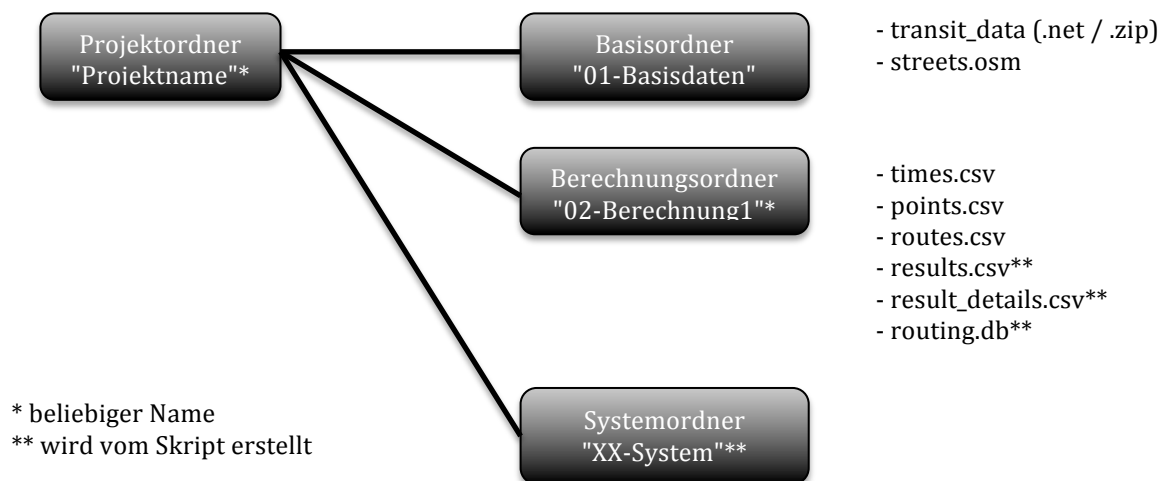
Um einen Feed grafisch visualisiert anzeigen zu lassen steht das Skript *scheduleviewer.py* zur Verfügung. Es wird wie folgt benutzt:

```
$ scheduleviewer.py <feed>
```

Feed ist der anzuzeigende Feed. Nach kurzem Laden, stellt das Skript unter der eingeblendeten Adresse lokal eine Webseite zur Verfügung, auf der der eingelesene Feed als Layer über Google Maps eingeblendet wird. Mit *Control + C* bzw. *Steuerung + C* kann das Skript beendet werden.

Berechnen von kürzesten Wegen

Um Berechnungen durchführen zu können, müssen die benötigten Daten in eine speziellen Ordnerstruktur eingefügt werden, sowie entsprechend umbenannt werden. Die Ordnerstruktur ist im folgenden Symbolisiert:



Als Ausgangsdaten werden benötigt:

- OpenStreetMap XML Daten den entsprechenden Gebiets
- Verkehrsdaten als GTFS-Feed oder Netzdatei
- times.csv, points.csv, routes.csv und config.txt, die im Abschnitt Dateiformate bzw. Konfigurationsdatei beschrieben werden

Innerhalb des Projektordners können sich beliebig viele Berechnungsordner befinden.

Die Berechnung selbst läuft wie folgt ab:

Nachdem man im Terminal in den entsprechenden Projektordner gewechselt ist, kann mit dem Aufruf

```
$ gst_process <Berechnungsordner>
```

die Berechnung gestartet werden. Bei der ersten Berechnung in einem Projektordner wird zusätzlich automatisch der Systemordner angelegt und entsprechende Datenbanken angelegt. Innerhalb eines Projektordners können sich beliebig viele Berechnungsordner befinden. Das Ergebnis der Berechnung ist in results.csv sowie result_details.csv gespeichert. Zusätzlich steht die sqlite-Datenbank *routing.db* zur Verfügung, die weitere Informationen enthält.

Dateiformate

Die Zeiten, Punkte und Routen müssen als CSV-Dateien vorliegen. Als Trennzeichen muss das Komma verwendet werden. Bei Zahlen, die ein Komma enthalten, muss diese durch eine Punkt ersetzt werden.

Exportdateien werden im gleichen Format erzeugt.

Tabellenform: (Spalten)

Zeiten:	- id:	Beliebige Ganzzahl.
	- start:	Start des Zeitfensters. (Format: TT:MM:JJJJ:HH:MM)
	- end:	Ende des Zeitfensters. (Format: TT:MM:JJJJ:HH:MM)
	- is_arrival_time:	0 oder 1. Gibt an, ob das Zeitfenster die Ankunft (1) oder die Abfahrt (0) definiert.
Punkte:	- id:	Beliebige Ganzzahl.
	- lat:	Latitude des Punktes. (WGS84)
	- lon:	Longitude des Punktes. (WGS84)
	- name:	beliebig
Routen:	- id:	Beliebige Ganzzahl.
	- origin:	id des Startpunktes.
	- destination:	id des Endpunktes.
	- time_id	id des Zeitfensters.

Konfigurationsdatei

Diese Datei enthält Name-Wert Paare, die durch das ‚=‘-Zeichen getrennt werden. Pro Zeile darf maximal ein Paar vorhanden sein.

Bsp:

```
link-dist=150
time-step=240
max-walk=1080
```

Mögliche Einstellungen:

Name	Wirkung und mögliche Werte
walking-speed	Die Geschwindigkeit, in der sich Fußgänger fortbewegen. In m/s. Punkt statt Komma verwenden!
time-step	Zeitabstand, in dem innerhalb der definierten Zeitfenster Berechnungen durchgeführt werden sollen. Je kleiner der Wert, desto genauer, aber auch zeitintensiver die Berechnungen.
max-walk	Entfernung in Metern, wie weit der max. Fußweg pro Berechnung sein darf.
walking-reluctance	Faktor um den der ‚virtuelle Fahrgast‘ lieber fährt als läuft. 1 – Fahren und Laufen wird als gleich lang empfunden. 2 – Laufen dauert gefühlt doppelt so lange.

Bekannte Fehler und Probleme

- Das Skript netToGtfs überprüft nicht, ob tatsächlich alle benötigten Information innerhalb der Netzdatei vorhanden sind.
- Beim internen Erstellen von kürzesten Wegen, entstehen teilweise negative Gewichtungen, die massenhaft als Fehlermeldung ausgegeben werden. Diese können einfach ignoriert werden.
- Falls zwischen zwei angebenen Punkten kein Weg existiert, wird zwar eine Fehlermeldung ausgegeben, aber kein Eintrag in die Ergebnisse geschrieben.
- Teilweise unverständliche Fehlermeldungen beim Aufruf von *gst_process*.