

Athens Open Trip Planner

Setup Tutorial

Εχουμε βασιστει σε αυτες τις πληροφοριες:

<http://docs.opentripplanner.org/en/latest/Basic-Usage/>

Σε μηχάνημα με Ubuntu 15.04 (λογικά λειτουργεί σε οποιαδήποτε έκδοση) και Oracle Java 1.7

Κατεβάζουμε το Open Street Map BPF file για Αθήνα

wget https://s3.amazonaws.com/metro-extracts.mapzen.com/athens_greece.osm.pbf

και ένα σωστό* GTFS του ΟΑΣΑ

*= Το αρχείο απο εδώ:

http://geodata.gov.gr/geodata/index.php?option=com_sobi2&sobi2Task=sobi2Details&catid=29&sobi2Id=190&Itemid=

Σύμφωνα με το https://developers.google.com/transit/gtfs/reference#stop_times_fields έχει ελλείψεις και δεν προχωράει η διαδικασία του graph build.

Τα αρχεία που μας δόθηκαν στον διαγωνισμό:

https://crowdpolicymaps.cartodb.com/u/crowdpolicy/tables/oasa_trips/public

https://crowdpolicymaps.cartodb.com/u/crowdpolicy/tables/oasa_stop_times/public

https://crowdpolicymaps.cartodb.com/u/atademo/tables/oasa_stops/public

https://crowdpolicymaps.cartodb.com/u/atademo/tables/oasa_shapes/public

https://crowdpolicymaps.cartodb.com/u/atademo/tables/oasa_routes/public

https://crowdpolicymaps.cartodb.com/u/atademo/tables/oasa_feed_info/public

https://crowdpolicymaps.cartodb.com/u/atademo/tables/oasa_calendar/public

https://crowdpolicymaps.cartodb.com/u/atademo/tables/oasa_calendar_dates/public

https://crowdpolicymaps.cartodb.com/u/atademo/tables/oasa_agency/public

έχουν περίττα columns, που πρόσθεσε το cartodb, αλλά δεν νομίζω να δημιουργούν κάποιο πρόβλημα, εκτός την αύξηση του μεγέθους. Μετά από rename του extension από .csv σε txt, αφαίρεση oasa_ prefix, και ξανά zip, ολοκληρώνεται η διαδικασία build του OTP graph, αλλά παρόλα αυτά δεν φέρνει λύσεις στο OTP οπότε πάλι κάποιο πρόβλημα υπάρχει και σε αυτά τα δεδομένα. Τελικά το graph build δουλέψε σωστά με ένα παλαιότερο GTFS που βρήκαμε.

Μεταφέρουμε το GTFS zip στο σωστό directory

```
mv google_transit_2012_03_08_fixed.zip otp/graph/athens
```

κατεβάζουμε το OTP jar

wget <http://dev.opentripplanner.org/jars/otp-0.18.0.jar>

```
mkdir -p otp/graph/athens
```

```
mv athens-gtfs.zip otp/graph/athens/
```

```
mv athens_greece.osm.pbf otp/graph/athens/
```

Φτιάχνουμε το graph:

```
java -Xmx2G -jar otp-0.18.0.jar --build otp/graph/athens/
```

και σηκώνουμε τον σερβερ:

```
java -Xmx2G -jar otp-0.18.0.jar --router athens --server --graphs otp/graph
```

<http://localhost:8080>

και ο Journey planner είναι έτοιμος, θα πρέπει να δίνει λύσεις

WiFi hotspot (RPi)

Η αναπτυξη έχει βασιστεί στο piratebox

http://www.piratebox.cc/raspberry_pi:diy

-----άλλες ιδέες για το Wifi Hot-Spot -----

openstreetmap OFFLINE

<http://forum.piratebox.cc/read.php?16,6988>

http://wiki.openstreetmap.org/wiki/OpenLayers_Local_Tiles_Example

RPi FM radio

<https://github.com/SchoolofArtsGent/Pirate-Radius>

http://www.icrobotics.co.uk/wiki/index.php/Turning_the_Raspberry_Pi_Into_an_FM_Transmitter

<http://www.instructables.com/id/Raspberry-Pi-Radio-Transmitter/>

----- Άλλες ιδέες για Journey planner -----

(Στησαμε και το **oneBusAway** αλλά δεν μας φάνηκε τόσο χρήσιμο τελικά στο hackathon, οπότε δεν το χρησιμοποιήσαμε. Κρατήσαμε όμως σημειώσεις :-)

Ξεκινήσαμε με αυτό το guide:

<https://github.com/OneBusAway/onebusaway-application-modules/wiki/OneBusAway-Quickstart-Guide>

στοχος είναι να πετύχουμε αυτό: (vanilla flavoured/deployed)

<http://tampa.onebusaway.org/>:

[http://tampa.onebusaway.org/where/standard/#m\(query\)q\(florida\)qll\(27.9769,-82.4459\)_0](http://tampa.onebusaway.org/where/standard/#m(query)q(florida)qll(27.9769,-82.4459)_0)

<http://transitfeeds.com/p/hillsborough-area-regional-transit/228>

Δυστυχώς το [ελληνικο GTFS](#) δεν είναι σε κάποιο από τα γνωστά directory:

<http://www.gtfs-data-exchange.com/>

<https://code.google.com/p/googletransitdatafeed/wiki/PublicFeeds>

<http://transitfeeds.com/p/athens-public-transit/298>

και αυτό που είναι διαθέσιμο έχει πρόβλημα και δεν παρσαρετέ αυτομάτα.

<https://developers.google.com/transit/gtfs/reference>

σηκώνουμε τον server GUI έτσι:

```
java -XX:-UseGCOverheadLimit -Xmx1G -server -jar
```

```
onebusaway-quickstart-assembly-1.1.13-webapp.war
```

Αυτό θα είχε ενδιαφέρον να το επεκτούμε για GTFS-RT output:

Simulation using the entire GTFS.

<https://github.com/vasile/transit-map>

<https://github.com/vasile/GTFS-viz>

hardcoded simulation:

<https://developers.google.com/transit/gtfs-realtime/gtfs-realtime-proto>

<https://gist.github.com/kurtraschke/11359441>

Bike option to explore 1)

add bike-rental system with realtime display and/or include in solution ability:

<http://docs.opentripplanner.org/en/latest/Configuration/#real-time-data>

<https://github.com/opentripplanner/OpenTripPlanner/wiki/Bike-Rental>

<https://groups.google.com/forum/#!topic/opentripplanner-dev/VALrOve0Gc4>

<https://sites.google.com/site/otpnotes/>

<http://api.citybik.es/>

<http://api.citybik.es/v2/>

<https://github.com/eskerda/PyBikes>

<https://github.com/f8full/open-bixi-gpstracker>

i.e we setup a proxy-server which receives android-app feed and submits bixi-xml to OTP

Bus option to explore 2)

a. add GTFS-RT custom server with **dummy** data (e.g. a bus going around)

<https://developers.google.com/transit/gtfs-realtime/gtfs-realtime-proto>

<https://gist.github.com/kurtraschke/11359441>

(or b.) use full blown GTFS **simulator** like server:

<https://github.com/vasile/transit-map>

<https://github.com/vasile/GTFS-viz>

- c. create android app for driver which submits lat-lon
- d. patch GTFS server (dummy/simulator) to submit FOR ONE BUS location using android app or arduino hardware
- e. add <http://dev.hsl.fi/navigator-proto/> user interface,

Background research

Technical references:

<https://github.com/OneBusAway/onebusaway-gtfs-realtime-visualizer/wiki>

Έξυπνη Στάση στην Αθήνα

<http://www.aftodioikisi.gr/proto-thema/g-dimitriadis-oi-exipnes-staseis-allazoun-tin-kathimerin-otita-ton-athinaion-dimiourgoume-ameses-theseis-ergasias-meso-jessica>

http://www.tsantiri.gr/ellada/ypografike_i_sumbasi_gia_tis_exupnes_staseis_leoforeion_-epe-ndusi_33_ekat_-euro/

SIRI-AVL

<https://kurtraschke.com/2011/04/open-standards-for-avl>

SIRI java API implementation

<https://github.com/CUTR-at-USF/SiriRestClient/wiki/Parsing-JSON-and-XML-on-Android>

Center for Urban Transportation (they forked OBA, OTP and more)

<https://github.com/CUTR-at-USF>

OBDII

<https://github.com/pires/android-obd-reader>

OBA

Operational in 5 cities in US and Canada, serving over 8 mil. end-users daily.

Outcomes of real-time information:

- 1,7% increase in weekday ridership in NYC
- 92% of users reporting increased satisfaction in Seattle
- increased number of transit-trips
- increased feeling of safety
- increase in walking
- real-time information leads to alignment of *actual* waiting time with *perceived* waiting time