

# Description of buses and trams localisation data

Marcin Luckner<sup>1</sup>, Karolina Kwasiborska, and Tomasz Zaremba

Faculty of Mathematics and Information Science, Warsaw University of Technology,  
Koszykowa 75, 00-662 Warsaw  
mluckner@mini.pw.edu.pl,  
WWW home page: <http://www.mini.pw.edu.pl/> lucknerm

**Abstract.** Description of bus and tram data extracted from Hive.

## 1 Localisation data

### 1.1 versionID

String Date of the last modification of the data structure in YYYYMMDD format.

### 1.2 line

String A line number.

### 1.3 brigade

String A number identifying a specific vehicle on given line. Combination of line and brigade numbers forms a unique identifier of a vehicle at certain point in time.

### 1.4 time

Datetime Timestamp of vehicle's GPS location measurement, obtained from the Warsaw API.

### 1.5 lon

float Vehicle's GPS longitude after noise removal.

### 1.6 lat

float Vehicle's GPS latitude after noise removal.

### 1.7 rawLon

float Vehicle's original GPS longitude.

### 1.8 rawLat

float Vehicle's original GPS latitude.

### 1.9 status

String A status determines whether a tram is STOPPED (did not move by at least 1 meter since the last record), MOVING\_SLOWLY (moved by 1-10 meters) or MOVING (moved by more than 10 meters). The initial status is set to UNKNOWN.

### 1.10 delay

Float - A delay in second including the delay between stops.

### 1.11 courseDirection

String A course direction stop name (the last stop from the timetable).

### 1.12 timetableIdentifier

String An identifier of the currently assigned timetable (format: concatenated scheduled stop time at the first stop and scheduled stop time at the last stop in the timetable e.g. 03:42:00-04:16:00).

### 1.13 timetableStatus

Enumeration A status of the assignment with a timetable.

- SAFE standard status set when we might be almost sure that we correctly assigned a timetable to the vehicle.
- UNSAFE status indicating that assigned timetable and all computed information might be incorrect.
- MISSING no timetable assigned.

### 1.14 receivedTime

Datetime A VM time of getting the data from API

### 1.15 processingFinishedTime

Datetime A VM time of writing the data to HDFS

### 1.16 onWayToDepot

Boolean A flag determines whether the vehicle is on the way to a depot set to true if its target stop name starts with ZET (tram depot) or ZEA (bus depot).

**1.17 overlapsWithNextBrigade**

Boolean A flag determines if the vehicle is so delayed that it seems to be moving with accordance to the timetable of the next vehicle of the same line on this route.

Example: two vehicles (A and B) of the same line should appear at some stop at 8:30 and 8:45 respectively. The first one is delayed by 15 minutes so a person waiting at the stop might think that B arrived at 8:45 and A did not show up at all.

**1.18 overlapsWithNextBrigadeStopLineBrigade**

String identifier of a vehicle (in line-brigade format) with accordance to whose timetable the vehicle described by the record is moving. Set only if overlapsWithNextBrigade is set to true, otherwise empty (see above).

**1.19 atStop**

Boolean A flag determines whether the vehicle is at the stop - set to true when the vehicle is less than 30 meters away from the stop. The distance is calculated based on the GPS coordinates using the haversine formula ([https://en.wikipedia.org/wiki/Haversine\\_formula](https://en.wikipedia.org/wiki/Haversine_formula)).

**1.20 speed**

Float The average speed of the vehicle calculated between two following locations (travelled distance divided by time). Empty for the first record.

**1.21 oldDelay**

Float A difference in seconds between planned time of arrival at a particular stop and actual arrival time, based on brigade+line+iteration matching, where iteration is the iteration of passing the route. Checked when a vehicle is less than 30 meters away from a stop. If the distance is greater than 30 meters, delay at the previous stop is displayed.

**1.22 serverId**

String identifier of the server collecting the data. Set by server.id parameter in the configuration file. Set to WUT if server.id is missing from the configuration.

**1.23 delayAtStopStopSequence**

Integer sequence number of the stop for which vehicles delay is measured.

### **1.24 previousStopStopSequence**

Integer - sequence number of the last visited stop on the route.

Stop sequence number - number of the stop in the sequence of stops on given route (taken directly from GTFS timetables). Typically the first stop on the route gets value 1 and the last one gets the value equal to the number of stops on the route.

### **1.25 nextStopStopSequence**

Integer - sequence number of the next stop on the route.

### **1.26 delayAtStopStopId**

String ID of the stop for which vehicles delay is measured.

Stop ID full identifier of the stop in the following form: [Stop ID from GTFS + - + stops peg number from GTFS], e.g. 3002\_01.

### **1.27 previousStopStopId**

String ID of the last visited stop on the route.

### **1.28 nextStopStopId**

String ID of the next stop on the route.

### **1.29 courseDirectionStopStopId**

String ID of the last stop on the route.

### **1.30 partition**

String A name of Hive partition.

## **2 Delays statistics**

### **2.1 window time**

datetime window timestamp

### **2.2 too early group count**

int number of vehicles which reached the stop too early.

**2.3 on time group count**

int number of vehicles which reached the stop on time.

**2.4 small delay group count**

int number of vehicles which reached the stop with a small delay.

**2.5 big delay group count**

int number of vehicles which reached the stop with a big delay.

**2.6 enormous delay group count**

int number of vehicles which reached the stop with an enormous delay.

**2.7 Summer categories**

- too early vehicles whose delay value is in the range  $(-\infty, -1)$
- on time vehicles whose delay is in the range  $< -1, 3 >$
- small delay vehicles whose delay is in the range  $(3, 6 >$
- big delay vehicles whose delay is in the range  $(6, 9 >$
- enormous delay vehicles whose delay is in the range  $(9, +\infty)$

**3 Group of vehicles****3.1 window time**

datetime window timestamp

**3.2 stop**

string - nearest/next stop based on which vehicles were grouped

**3.3 depot flag**

boolean flag which indicates whether the group is in the depot or not

- true when more than half of vehicles is in the depot;
- false otherwise

**3.4 vehicle time**

datetime timestamp of the vehicle record

### **3.5 vehicle key**

string identifier of the vehicle (line-brigade key)

### **3.6 longitude**

float longitude GPS

### **3.7 latitude**

float latitude GPS

### **3.8 moving status**

string - determines whether a vehicle is STOPPED (did not move by at least 1 meter since the last record), MOVING\_SLOWLY (moved by 1-10 meters) or MOVING (moved by more than 10 meters).