

Waze Data Feed

Specification Document

Version 2.9

Revision History

The following revision table presents the changes made for each version of this document.

| Revision | Date | State/Description |
|----------|---------------|--|
| 1.5 | February 2014 | Creation based on GeoRSS API spec-doc (Chen Barshai) |
| 2.0 | August 2014 | Updated (Noam Reshef) |
| 2.5 | November 2014 | Updated - removed fields, separated JSON and XML tables, added reliability score instead of confidence |
| 2.6 | November 2014 | updates every 2 min |
| 2.7 | November 2015 | Unusual events, New url, new examples |
| 2.7.1 | May 2016 | Updated-confidence and reliability scores |
| 2.7.2 | June 2016 | Updated linqmap:speed units |
| 2.8 | March 2017 | Added configuration options, and specified irregularity parameters |
| 2.8 | August 2017 | Added broken traffic light alert, receive number of thumbs up by users, get image of irregularity |
| 2.9 | August 2022 | Adjust to PartnerHub new platform |

Table of Contents

[Overview](#)

[Terms and Acronyms](#)

[Retrieving Waze traffic-data](#)

[Using Waze traffic-data](#)

[Traffic Alerts](#)

[Reliability](#)

[Traffic Alerts - example](#)

[JSON format](#)

[XML format](#)

[Alert Types](#)

[Traffic Jams information](#)

[Traffic jam results - example](#)

[JSON Format](#)

[XML Format](#)

[Appendix](#)

[Road types table](#)

Introduction

Overview

This document details the structure and content of the Waze real-time traffic data feed.

Waze maps provide information about specific routes to assist motorists in avoiding traffic jams. Our maps and traffic are dynamically updated by a continually expanding network of drivers worldwide. This, of course, reduces the importance of predefined routes. The most accurate representations are through poly-lines – where the traffic is geographically-based, independent of specific map attributes.

We provide information about traffic jams and events that affect road conditions, either from drivers using Waze, a.k.a. Wazers, or from external sources. Wazers provide road reports, or UGIs (User-Generated Incidents), which include detailed information about specific traffic and road conditions. They may issue reports from the location at which they are currently located or, if no longer at the location, within 30 minutes after the event occurred. We are also able to provide automatic alerts for what we call Unusual Traffic - incidents that affect a large number of users and fall outside the normal traffic patterns for a given day and time.

The time period that affects the results can be internally adjusted by the Waze server configuration to provide best results, based upon local conditions. These include weather conditions, a car stopped at the shoulder, etc.

Terms and Acronyms

The following table provides a list of terms and acronyms used in this document.

| Term or Acronym | Definition |
|-----------------|--|
| GPS | Global Positioning System |
| TMC | Traffic Message Channel |
| UGI | User-Generated Incident |
| Wazer | An active member of the Waze community |
| Wazer level | 1-6, 1 being the entry level - indicating a user's level of experience and trustworthiness |
| WGS 84 | World Geodetic System (1984) |
| GeoRSS | A standard for encoding location as part of a Web feed . |
| JSON/XML | A format used to transmit data between a server and web application |

Retrieving Waze traffic-data

Waze uses the GeorSS API to internally-retrieve traffic data.

We provide “Waze for Cities” partners with an XML/JSON file containing all the data approved for sharing, as per the partner agreement.

The feed is provided by URL.

- **Feed content per managed area:** The feed will include all available data per managed area the partner was approved for. To check the partner current approved managed areas or to request for another/change existing one, use the *Settings* section on the PartnerHub.
- **Base URL for each managed area:** To get the URL for the feed associated with a managed area, go to the *Toolbox* section, then choose *Waze Data Feed*, and then check the *Feed Links* section on the table. Choose JSON/XML format to get the url and current feed content. The base url will have this structure:

```
https://www.waze.com/partnerhub-api/waze-feed-access-token/<unique-token>?format=<format>
```

- **Adjustments to the feed content:** after getting the base feed URL, you will be able to customize it by adding additional parameters. These are the additional parameters available:

| Parameter | Possible values | Details and example |
|-----------|--|--|
| types | alerts/ irregularities/ traffic | Filter the type of information. Default is to fetch all types, adding this filter will decrease response size and will make server response faster. Example: <code>https://www.waze.com/partnerhub-api/waze-feed-access-token/<unique-token>?format=<format>&types=alerts,traffic</code> |
| acotu | false/true | If true adds a field “nThumbsUp” to alerts which have been “thumbed up” by Wazers. Default to false. Setting to true makes server response slower. Example: <code>https://www.waze.com/partnerhub-api/waze-feed-access-token/<unique-token>?format=<format>&acotu=true</code> |
| atf | Alert types: See here | Filters specific alerts types. Example: <code>https://www.waze.com/partnerhub-api/waze-feed-access-token/<unique-token>?format=<format>&atf=ROAD_CLOSED,</code> |

| | | |
|-----|------------|---|
| | | ACCIDENT |
| fa | false/true | <p>If true includes alerts which have been reported by a partner feed, in addition to alerts reported by Wazers. If false, output includes only alerts reported by users. Default to true. Example:</p> <p><code>https://www.waze.com/partnerhub-api/waze-feed-access-token/<unique-token>?format=<format>&fa=true</code></p> |
| ofa | false/true | <p>If true includes only alerts which have been reported via partner feed. If false includes alerts which have been reported by partner feed and Wazers via the application. Default to false. Example:</p> <p><code>https://www.waze.com/partnerhub-api/waze-feed-access-token/<unique-token>?format=<format>&ofa=true</code></p> |

Using Waze traffic data

Waze traffic data consists of the following information:

1. *General information*: time-stamp of the file, geographic area from which the data was retrieved, etc.
2. *Traffic alerts*: traffic incidents reported by users.
3. *Traffic jams*: traffic slowdown information generated by the service based on a user's location and speed.
4. *Unusual Traffic (Irregularities)*: alerts and traffic jams that affect an exceptionally large number of users.

Note: From time-to-time new fields may be added to the feed. Once those fields are permanently available this document will be modified and redistributed to the relevant partners.

Traffic Alerts

The 'Alerts' section includes all traffic data reported by Waze users through the Waze mobile application.

Reliability

Each alert gets a **reliability** score based on other user's reactions ('Thumbs up', 'Not there' etc.) and the level of the reporter (Wazers gain levels by contributing to the map, starting at level 1 and reaching up to level 6. The higher the level, the more experienced and trustworthy the Wazer.) The score (0-10) indicates how reliable the report is.

Confidence

Each alert gets a **confidence** score based on other user's reactions ('Thumbs up', 'Not there'). The score ranges between 0 and 10. A higher score indicates more positive feedback from Waze users.

The data include an array of alerts with the following attributes:

XML

Specification Document for Real-time Traffic Data

| Element | Value | Description |
|----------------------------------|---------------------------|--|
| pubDate | Time | Publication date |
| georss:point | Coordinates | Location per report (Lat long) |
| linqmap:uuid | String | Unique system ID |
| linqmap:magvar | Integer (0-359) | Event direction (Driver heading at report time. 0 degrees at North, according to the driver's device) |
| linqmap:type | See alert type table | Event type |
| linqmap:subtype | See alert sub types table | Event sub type - depends on atof parameter |
| linqmap:reportDescription | String | Report description (supplied when available) |
| linqmap:street | String | Street name (as is written in database, no canonical form, may be null) |
| linqmap:city | String | City and state name [City, State] in case both are available, [State] if not associated with a city. (supplied when available) |
| linqmap:country | String | (see two letters codes in http://en.wikipedia.org/wiki/ISO_3166-1) |
| linqmap:roadType | Integer | Road type (see road types table in the appendix) |
| linqmap:reportRating | Integer | User rank between 1-6 (6 = high ranked user) |
| linqmap:jamUuid | string | If the alert is connected to a jam - jam ID |
| linqmap:Reliability (new) | 0-10 | How reliable is the report, 10 being most reliable. Based on reporter level and user responses |
| linqmap:reportByMunicipalityUser | Boolean | Alert reported by municipality user (partner) Optional. |
| linqmap:nThumbsUp | integer | Number of thumbs up by users |

JSON

Specification Document for Real-time Traffic Data

| Element | Value | Description |
|--------------------------|---------------------------|--|
| pubMillis | Timestamp | Publication date (Unix time – milliseconds since epoch) |
| location | Coordinates | Location per report (X Y - Long-lat) |
| uuid | String | Unique system ID |
| magvar | Integer (0-359) | Event direction (Driver heading at report time. 0 degrees at North, according to the driver's device) |
| type | See alert type table | Event type |
| subtype | See alert sub types table | Event sub type - depends on atof parameter |
| reportDescription | String | Report description (supplied when available) |
| street | String | Street name (as is written in database, no canonical form, may be null) |
| city | String | City and state name [City, State] in case both are available, [State] if not associated with a city. (supplied when available) |
| country | String | (see two letters codes in http://en.wikipedia.org/wiki/ISO_3166-1) |
| roadType | Integer | Road type (see road types table in the appendix) |
| reportRating | Integer | User rank between 1-6 (6 = high ranked user) |
| jamUuid | string | If the alert is connected to a jam - jam ID |
| Reliability (new) | 0-10 | How reliable is the report..... |
| reportByMunicipalityUser | Boolean | Alert reported by municipality user (partner) Optional. |
| nThumbsUp | integer | Number of thumbs up by users |

Traffic Alerts - example*XML format*

```

<item>
  <pubDate>Thu Nov 26 14:02:29 +0000 2015</pubDate>
  <georss:point>45.02395420471421 7.670893079148089</georss:point>
  <linqmap:uuid>9fd1ee98-7b56-37e9-a2d4-72e9478dd838</linqmap:uuid>
  <linqmap:magvar>6</linqmap:magvar>
  <linqmap:type>WEATHERHAZARD</linqmap:type>
  <linqmap:subtype>HAZARD_ON_ROAD_CONSTRUCTION</linqmap:subtype>
  <linqmap:reportDescription>
    scambio di carreggiata causa lavori dalle 00:00 del 16 novembre 2015 alle 23:59 del 21
    gennaio 2016
  </linqmap:reportDescription>
  <linqmap:city>Torino</linqmap:city>
  <linqmap:country>IT</linqmap:country>
  <linqmap:roadType>4</linqmap:roadType>
  <linqmap:reportRating>0</linqmap:reportRating>
  <linqmap:reliability>10</linqmap:reliability>
</item>
<item>
  <pubDate>Thu Nov 26 14:02:26 +0000 2015</pubDate>
  <georss:point>45.02395420471421 7.670893079148089</georss:point>
  <linqmap:uuid>ed06a695-53ee-347c-a6eb-133bf8746880</linqmap:uuid>
  <linqmap:magvar>6</linqmap:magvar>
  <linqmap:type>WEATHERHAZARD</linqmap:type>
  <linqmap:subtype>HAZARD_ON_ROAD_CONSTRUCTION</linqmap:subtype>
  <linqmap:reportDescription>
    chiusura notturna causa lavori di manutenzione dalle 23:00 alle 05:30, solo nei giorni
    feriali dalle 23:00 del 9 novembre 2015 alle 05:30 del 5 dicembre 2015
  </linqmap:reportDescription>
  <linqmap:city>Torino</linqmap:city>
  <linqmap:country>IT</linqmap:country>
  <linqmap:roadType>4</linqmap:roadType>
  <linqmap:reportRating>0</linqmap:reportRating>
  <linqmap:reliability>7</linqmap:reliability>
</item>

```

JSON format

```
{ "country": "IT", "roadType": 1, "magvar": 258, "subtype": "", "reportRating": 0, "reliability": 6, "reportDescription": "blocco del traffico per alcuni veicoli nella ZTL (Zona Traffico Limitato) Non possono circolare Veicoli per il trasporto persone Dal lunedì al venerdì, dalle ore 8 alle ore 19 - veicoli benzina Eur", "location": { "x": 7.6800935614336545, "y": 44.9991565694201 }, "type": "WEATHERHAZARD", "uuid": "39d9dc07-bd74-3b35-ba6b-833f5cbd1ce1", "pubMillis": 1448546704610 }, { "country": "IT", "magvar": 0, "subtype": "ROAD_CLOSED_EVENT", "city": "Nichelino", "street": "Via Fenestrelle", "reportRating": 0, "reliability": 9, "reportDescription": "lavori", "location": { "x": 7.627331910061528, "y": 45.00419885851123 }, "type": "ROAD_CLOSED", "uuid": "1064e72c-0d3b-332d-95c6-1dcab524aa5c", "pubMillis": 1446918728242 },
```

Alert Types

Waze currently supports the following types and subtypes of user-generated alerts:

| ID | Alert type | Alert Subtype | |
|----|------------------------|---|--|
| | ACCIDENT | <ul style="list-style-type: none"> • ACCIDENT_MINOR • ACCIDENT_MAJOR • NO_SUBTYPE | |
| | JAM | <ul style="list-style-type: none"> • JAM_MODERATE_TRAFFIC • JAM_HEAVY_TRAFFIC • JAM_STAND_STILL_TRAFFIC • JAM_LIGHT_TRAFFIC • NO_SUBTYPE | |
| | WEATHERHAZARD / HAZARD | <ul style="list-style-type: none"> • HAZARD_ON_ROAD • HAZARD_ON_SHOULDER • HAZARD_WEATHER • HAZARD_ON_ROAD_OBJECT • HAZARD_ON_ROAD_POT_HOLE • HAZARD_ON_ROAD_ROAD_KILL • HAZARD_ON_SHOULDER_CAR_STOPPED • HAZARD_ON_SHOULDER_ANIMALS • HAZARD_ON_SHOULDER_MISSING_SIGN • HAZARD_WEATHER_FOG • HAZARD_WEATHER_HAIL • HAZARD_WEATHER_HEAVY_RAIN • HAZARD_WEATHER_HEAVY_SNOW • HAZARD_WEATHER_FLOOD • HAZARD_WEATHER_MONSOON • HAZARD_WEATHER_TORNADO • HAZARD_WEATHER_HEAT_WAVE • HAZARD_WEATHER_HURRICANE • HAZARD_WEATHER_FREEZING_RAIN • HAZARD_ON_ROAD_LANE_CLOSED • HAZARD_ON_ROAD_OIL • HAZARD_ON_ROAD_ICE • HAZARD_ON_ROAD_CONSTRUCTION • HAZARD_ON_ROAD_CAR_STOPPED • HAZARD_ON_ROAD_TRAFFIC_LIGHT_FAULT • NO_SUBTYPE | |
| | MISC | <ul style="list-style-type: none"> • NO_SUBTYPE | |
| | CONSTRUCTION | <ul style="list-style-type: none"> • NO_SUBTYPE | |
| | ROAD_CLOSED | <ul style="list-style-type: none"> • ROAD_CLOSED_HAZARD • ROAD_CLOSED_CONSTRUCTION • ROAD_CLOSED_EVENT • NO_SUBTYPE | |

Traffic Jams information

The jams section includes data gathered in real time about traffic slowdowns on specific road segments.

The URL includes an array of jams and the start time and date of the report. Each jam includes the parameters detailed below.

Waze generates traffic jam information by processing the following data sources:

- GPS location-points sent from user phones (users who drive while using the app) and calculations of the current average speed vs. free-flow speed (maximum speed measured on the road-segment). For Unusual traffic (irregularities) Waze uses historic average speeds (on 30 minute time-slots).
- User generated reports - reports shared by Waze users who encounter traffic jams. These appear as regular alerts, and also affect the way we identify and present traffic jams.

Traffic jam parameters are described in the following table:

XML

| Element | Value | Description |
|---------------------------|--|--|
| pubDate | Time | Publication date |
| linqmap:type | String | TRAFFIC_JAM |
| georss:line | List of Longitude and Latitude coordinates | Traffic jam line string (supplied when available) |
| linqmap:speed | Float | Current average speed on jammed segments in meters/seconds |
| linqmap:speedKPH | Float | Current average speed on jammed segments in KM/h |
| linqmap:length | Integer | Jam length in meters |
| linqmap:delay | Integer | Delay of jam compared to free flow speed, in seconds (in case of block, -1) |
| linqmap:street | String | Street name (as is written in database, no canonical form. (supplied when available) |
| linqmap:city | String | City and state name [City, State] in case both are available, [State] if not associated with a city (supplied when available) |
| linqmap:country | String | available on EU (world) server (see two letters codes in http://en.wikipedia.org/wiki/ISO_3166-1) |
| linqmap:roadType | Integer | Road type (see road types table in the appendix) |
| linqmap:startNode | String | Nearest Junction/steet/city to jam start (supplied when available) |
| linqmap:endNode | String | Nearest Junction/steet/city to jam end (supplied when available) |
| linqmap:level | 0 - 5 | Traffic congestion level (0 = free flow 5 = blocked). |
| linqmap:uuid | Long integer | Unique jam ID |
| linqmap:turnLine | Coordinates | A set of coordinates of a turn - only when the jam is in a turn (supplied when available) |
| linqmap:turnType | String | What kind of turn is it - left, right, exit R or L, continue straight or NONE (no info) (supplied when available) |
| linqmap:blockingAlertUuid | string | if the jam is connected to a block (see alerts) |

JSON

| Element | Value | Description |
|-------------------|--|--|
| pubMillis | Timestamp | Publication date (Unix time – milliseconds since epoch) |
| type | String | TRAFFIC_JAM |
| line | List of Longitude and Latitude coordinates | Traffic jam line string (supplied when available) |
| speed | Float | Current average speed on jammed segments in meters/seconds |
| speedKPH | Float | Current average speed on jammed segments in Km/h |
| length | Integer | Jam length in meters |
| delay | Integer | Delay of jam compared to free flow speed, in seconds (in case of block, -1) |
| street | String | Street name (as is written in database, no canonical form. (supplied when available) |
| city | String | City and state name [City, State] in case both are available, [State] if not associated with a city. (supplied when available) |
| country | String | available on EU (world) server (see two letters codes in http://en.wikipedia.org/wiki/ISO_3166-1) |
| roadType | Integer | Road type (see road types table in the appendix) |
| startNode | String | Nearest Junction/steet/city to jam start (supplied when available) |
| endNode | String | Nearest Junction/steet/city to jam end (supplied when available) |
| level | 0 - 5 | Traffic congestion level (0 = free flow 5 = blocked). |
| uuid | Long integer | Unique jam ID |
| turnLine | Coordinates | A set of coordinates of a turn - only when the jam is in a turn (supplied when available) |
| turnType | String | What kind of turn is it - left, right, exit R or L, continue straight or NONE (no info) (supplied when available) |
| blockingAlertUuid | string | if the jam is connected to a block (see alerts) |

Traffic jam results - example*JSON Format*

```
{
  "jams": [
    {
      "country": "US",
      "city": "Englewood, NJ",
      "level": 5,
      "line": [
        { "x": -73.980907, "y": 40.885657 },
        { "x": -73.9803, "y": 40.885302 }
      ],
      "speedKMH": 0,
      "length": 65,
      "turnType": "NONE",
      "type": "NONE",
      "uuid": 1320005294,
      "endNode": "S Dean St",
      "speed": 0,
      "segments": [
        {}
      ],
      "blockingAlertUuid": "9fd8bf93-0e69-46c3-8447-270181a98b93",
      "roadType": 7,
      "delay": -1,
      "street": "E Forest Ave",
      "id": 1320005294,
      "pubMillis": 1659939026804
    }
  ]
}
```


XML Format

```
<item>
<title>jam</title>
<pubDate>Mon Aug 8 06:10:26 +0000 2022</pubDate>
<linqmap:id>1320005294</linqmap:id>
<linqmap:uuid>1320005294</linqmap:uuid>
<linqmap:type>NONE</linqmap:type>
<georss:line>40.885657 -73.980907 40.885302 -73.9803</georss:line>
<linqmap:speed>0.0</linqmap:speed>
<linqmap:speedKMH>0.0</linqmap:speedKMH>
<linqmap:length>65</linqmap:length>
<linqmap:delay>-1</linqmap:delay>
<linqmap:endNode>S Dean St</linqmap:endNode>
<linqmap:street>E Forest Ave</linqmap:street>
<linqmap:city>Englewood, NJ</linqmap:city>
<linqmap:country>US</linqmap:country>
<linqmap:roadType>7</linqmap:roadType>
<linqmap:level>5</linqmap:level>
<linqmap:turnType>NONE</linqmap:turnType>
<linqmap:blockingAlertUuid>9fd8bf93-0e69-46c3-8447-270181a98b93</linqmap:blockingAlertUuid>
</item>
```

Irregularities (Unusual traffic jams)

These are traffic jams identified by the system as irregular by taking into account historical speed data. It includes the following attributes:

XML

| Element | Value | Description |
|----------------------|--|--|
| linqmap:id | Long integer | Irregularity Identifier |
| detectionDate | Date timestamp | Date of irregularity |
| detectionDateMillis | Unix date in milliseconds | Date of irregularity |
| updateDate | Date timestamp | Last update |
| updateDateMillis | Unix date in milliseconds | Last update |
| georss:line | List of Longitude and Latitude coordinates | Traffic jam line string (supplied when available) |
| linqmap:type | String | Irregularity type - NONE(0, "NONE"), SMALL(1, "Small"), MEDIUM(2, "Medium"), LARGE(3, "Large"), HUGE(4, "Huge"); |
| linqmap:speed | Float number | Traffic speed in irregularity |
| linqmap:regularSpeed | Float number | Historical regular speed in segment |
| linqmap:delaySeconds | Number | Delay in seconds from regular speed |
| linqmap:seconds | Number | Current traffic speed |
| linqmap:length | Number | Irregularity length |
| linqmap:trend | Number | -1 improving, 0 constant, 1 getting worse |
| linqmap:street | String | Street name |
| linqmap:city | String | City name |

JSON

| Element | Value | Description |
|---------------------|--|--|
| id | Long integer | Irregularity Identifier |
| detectionDate | Date timestamp | Date of irregularity |
| detectionDateMillis | Unix date in milliseconds | Date of irregularity |
| updateDate | Date timestamp | Last update |
| updateDateMillis | Unix date in milliseconds | Last update |
| line | List of Longitude and Latitude coordinates | Traffic jam line string (supplied when available) |
| type | String | Irregularity type - NONE(0, "NONE"), SMALL(1, "Small"), MEDIUM(2, "Medium"), LARGE(3, "Large"), HUGE(4, "Huge"); |
| speed | Float number | Traffic speed in irregularity |
| regularSpeed | Float number | Historical regular speed in segment |
| delaySeconds | Number | Delay in seconds from regular speed |
| seconds | Number | Current traffic speed |
| length | Number | Irregularity length |
| trend | Number | -1 improving, 0 constant, 1 getting worse |
| street | String | Street name |
| city | String | City name |
| country | String | Country name |
| severity | Float number | Calculated severity of irregularity 0-5 (5 = most severe) |
| jamLevel | Number | 1-4, 4 being worst jam level |
| driversCount | Number | Number of Wazers in irregularity |
| alertsCount | Number | How many alerts from Wazers in irregularity segments |

Irregularity example:

```

<title>irregularity</title>
<description>LineString consists of lat, lon pairs</description>
<guid>urn:uuid:d496f4e3-7fd8-4169-ac6208e36dffa7f0</guid>
<link>http://georss.org/example/simple/point</link>
<author>LinQmap</author>
<linqmap:id>120742770</linqmap:id>
<detectionDate>Wed Mar 15 12:22:18 +0000 2017</detectionDate>
<detectionDateMillis>1489580538138</detectionDateMillis>
<updateDate>Wed Mar 15 12:27:50 +0000 2017</updateDate>
<updateDateMillis>1489580870567</updateDateMillis>
<georss:line>
-33.3767 -70.656673 -33.377921 -70.65339 -33.379652 -70.648736 -33.37997 -70.647886 -33.38044
-70.646657 -33.381017 -70.64513 -33.381244 -70.644564 -33.381506 -70.643949 -33.382137
-70.642744 -33.382773 -70.641687 -33.383395 -70.640793 -33.383692 -70.640355 -33.384082
-70.639751 -33.38476 -70.638786 -33.38533 -70.638015
</georss:line>
<linqmap:type>Small</linqmap:type>
<linqmap:speed>13.82</linqmap:speed>
<linqmap:regularSpeed>22.0</linqmap:regularSpeed>
<linqmap:delaySeconds>355</linqmap:delaySeconds>
<linqmap:seconds>519</linqmap:seconds>
<linqmap:length>1995</linqmap:length>
<linqmap:trend>0</linqmap:trend>
<linqmap:endNode>Acceso Autopista Vespucio Norte</linqmap:endNode>
<linqmap:street>Av. Américo Vespucio</linqmap:street>
<linqmap:city>Recoleta</linqmap:city>
<linqmap:country>Cl</linqmap:country>
<linqmap:severity>1.493503420502874</linqmap:severity>
<linqmap:jamLevel>3</linqmap:jamLevel>

```

```
<linqmap:driversCount>335</linqmap:driversCount>
<linqmap:alertsCount>4</linqmap:alertsCount>
<linqmap:alerts>
<linqmap:alert>
<item>
<title>alert</title>
<pubDate>Wed Mar 15 12:18:23 +0000 2017</pubDate>
<georss:point>-33.377246 -70.655208</georss:point>
<linqmap:uuid>df2cf70e-f6e9-3f16-b64e-8fc440d81789</linqmap:uuid>
<linqmap:magvar>113</linqmap:magvar>
<linqmap:type>JAM</linqmap:type>
<linqmap:subtype>JAM_HEAVY_TRAFFIC</linqmap:subtype>
<linqmap:street>Av. Américo Vespucio</linqmap:street>
<linqmap:city>Recoleta</linqmap:city>
<linqmap:country>CL</linqmap:country>
<linqmap:roadType>2</linqmap:roadType>
```

Ruslan • 3 mins

Appendix

Road types table

| Value | Type |
|-------|------------------|
| 1 | Streets |
| 2 | Primary Street |
| 3 | Freeways |
| 4 | Ramps |
| 5 | Trails |
| 6 | Primary |
| 7 | Secondary |
| 8, 14 | 4X4 Trails |
| 15 | Ferry crossing |
| 9 | Walkway |
| 10 | Pedestrian |
| 11 | Exit |
| 16 | Stairway |
| 17 | Private road |
| 18 | Railroads |
| 19 | Runway/Taxiway |
| 20 | Parking lot road |
| 21 | Service road |