8-9-2018

Webinar: Meeting & Exceeding Mobility User Expectations with Real-Time Transit Information

Sean J. Barbeau  
*University of South Florida*

Derek Fretheim  
*Moovel*

Let us know how access to this document benefits you.

Follow this and additional works at: [https://pdxscholar.library.pdx.edu/trec_webinar](https://pdxscholar.library.pdx.edu/trec_webinar)

Part of the [Transportation Commons](https://pdxscholar.library.pdx.edu/trec_webinar), [Transportation Engineering Commons](https://pdxscholar.library.pdx.edu/trec_webinar), and the [Urban Studies Commons](https://pdxscholar.library.pdx.edu/trec_webinar)

**Recommended Citation**


[https://pdxscholar.library.pdx.edu/trec_webinar/31](https://pdxscholar.library.pdx.edu/trec_webinar/31)

This Book is brought to you for free and open access. It has been accepted for inclusion in TREC Webinar Series by an authorized administrator of PDXScholar. For more information, please contact pdxscholar@pdx.edu.
GTFS - realtime

Derek Fretheim, Director Business Development
August 9, 2018
moovel builds award-winning white labeled mobility apps
Mobility convergence

Transit agencies and cities are finding ways to provide mobility within their community.

User
- Seamless access to a whole range of mobility services
- Booking & payment included
- Preferences showing real-time availability

Transit Agencies
- Operational data
- Route optimization
- New markets needs
- Focus on Customer Experience

Cities
- Insights into different mobility patterns
- Data for planning purposes
- How to manage and operate the urban mobility network

MSPs
- Dynamic data showing vehicle locations and availability
- Different mobility options
Aggregating mobility in one app

Karlsruhe KVV Transit App - Introduced in 2015, our first white-label MaaS app to power “look, book and pay” functionality across modes of transportation with multiple providers.
Aggregating 24 providers in one app

FASTLink DTLA integrates 24 different transit providers in our Multimodal APP
Building a reliable Maas Solution with GTFS-RT

- GTFS-RT + ITS/real-time traffic data + MSP integration is challenging and complex
- Trip planning with MaaS requires reliable and accurately managed GTFS-RT data sources
- APP development using GTFS-RT requires data reliability, stability and standard formatting
- Customer satisfaction is ‘sticky’ when user preferences are aligned with real-time data
GTFS-realtime v2.0

Sean J. Barbeau, Ph.D.
Principal Mobile Software Architect for R&D
Center for Urban Transportation Research
University of South Florida
Why real-time transit info?

• Real-time information (RTI) has many benefits
  – Shorter perceived wait time [1][8]
  – Shorter actual wait time [1]
  – Lowers learning curve for new riders [2]
  – Increased feeling of safety (e.g., at night) [5][6]
  – Improved perception of agency [8]
  – Increased ridership [3][7]

• Improve the rider experience without increasing vehicle frequency

• Emerging analytics tools are powered by this data

• **GTFS-realtime** is becoming de facto standard
  – Over 50 agencies now have GTFS-realtime feeds!
Quality is important!

• In one study, **9% of riders** said they took the bus **less often** due to errors in RTI\[9\]

• GTFS-realtime v2.0 will help agencies produce better quality RTI

---

WHAT’S WRONG WITH GTFS-REALTIME V1.0?
Problem with GTFS-realtime v1.0

• GTFS-realtime includes:
  – Trip Updates (arrival predictions)
  – Vehicle Positions
  – Service Alerts

• A LOT of *optional* fields – 56 out of 63 (89%)

• Leads to sub-optimal feeds
  – Poor data quality
  – Bad rider experience
  – Inaccurate analytics – garbage in, garbage out
Example 1 – Vehicle Position

• All timestamps are optional
• When was position calculated?

```plaintext
header {
  gtfs_realtime_version: "1.0"
}
entity {
  id: "d131dd02"
  vehicle {
    position {
      latitude: 28.04265
      longitude: -82.45945
    }
  }
}
```
Example 2 – Loop route

```
trip {
  trip_id: "277725"
}
stop_time_update {
  arrival {
    delay: 900 // 15 minutes
  }
  stop_id: "A"
}
```

- `stop_sequence` field is `optional`
- Is 15 minute delay before or after `stop_id B`?
• ~69% of feeds (54) we analyzed had errors
Clarifying what’s really optional

GTFS-REALTIME V2.0
GTFS-realtime v2.0

• Defines new transit-specific requirements

• Each field is labeled as either:
  – Required
  – Optional
  – Conditionally required
    • See Description field for when this field is required
message StopTimeUpdate

Realtime update for arrival and/or departure events for a given stop on a trip. Please also refer to the general discussion of stop time updates in the TripDescriptor and trip updates entities documentation.

Updates can be supplied for both past and future events. The producer is allowed, although not required, to drop past events. The update is linked to a specific stop either through stop_sequence or stop_id, so one of these fields must necessarily be set. If the same stop_id is visited more than once in a trip, then stop_sequence should be provided in all StopTimeUpdates for that stop_id on that trip.

Fields

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Required</th>
<th>Cardinality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stop_sequence</td>
<td>uint32</td>
<td>Conditionally required</td>
<td>One</td>
<td>Must be the same as in stop_times.txt in the corresponding GTFS feed. Either stop_sequence or stop_id must be provided within a StopTimeUpdate - both fields cannot be empty. stop_sequence is required for trips that visit the same stop_id more than once (e.g., a loop) to disambiguate which stop the prediction is for.</td>
</tr>
<tr>
<td>stop_id</td>
<td>string</td>
<td>Conditionally required</td>
<td>One</td>
<td>Must be the same as in stops.txt in the corresponding GTFS feed. Either stop_sequence or stop_id must be provided within a StopTimeUpdate - both fields cannot be empty.</td>
</tr>
</tbody>
</table>
What’s new in GTFS-realtime v2.0

Better quality real-time information

Read more...

Quick analysis of feeds

GTFS-REALTIME
VALIDATOR
GTFS-realtime Validator

• Test your v1 and v2 feeds for errors
GTFS-rt Validator – View data & errors
Evaluation of industry feeds

• Created open-source tool to batch validate 78 out of 130 GTFS-realtime feeds catalogued on TransitFeeds.com

• 69% (54) feeds had errors, and 74% (58) had warnings
Most Frequent Errors and Warnings in GTFS-realtime feeds

Number of feeds with error/warning

- E011 - GTFS-rt stop_id does not exist in GTFS data
- E022 - Sequential stop_time_update times are not increasing
- E045 - GTFS-rt stop_time_update stop_sequence and stop_id do not match GTFS
- E023 - trip start_time does not match first GTFS arrival_time
- E041 - trip doesn’t have any stop_time_updates
- E012 - Header timestamp should be greater than or equal to all other timestamps
- E001 - Not in POSIX time
- W009 - schedule_relationship not populated
- W001 - timestamps not populated
- W002 - vehicle_id not populated
- W004 - vehicle speed is unrealistic
- W006 - trip_update missing trip_id

What’s next for Transit Agencies?

• Require that all AVL vendors provide GTFS-realtime v2.0 feeds (especially in RFPs)
  – Github is official new home of GTFS/GTFS-realtime specs - https://github.com/google/transit
• Run GTFS-realtime validator frequently
  – https://github.com/CUTR-at-USF/gtfs-realtime-validator
• Also require that vendors follow GTFS Best Practices
  – http://gtfs.org/best-practices/
• Communicate with other agencies and app developers
  – See https://github.com/CUTR-at-USF/awesome-transit#community for resources
What’s next for GTFS-rt Community?

• Create GTFS-realtime Best Practices
  – “Warning” from GTFS-realtime validator\(^1\)
  – Proposals without unanimous agreement
  – Other community input

• Clarify more GTFS-realtime gray areas
  – Either new proposals, or in best practices

• Better targeted documentation
  – More focused on use cases and features (e.g., ability to cancel trips)

What’s next for GTFS-rt Community?

• Continue to add new rules to GTFS-realtime Validator

• Hosting GTFS-realtime Validator as a service for agencies and vendors

• Tackle GTFS Services Changes use cases
  – Changes to network that happen frequently

• Develop additional open-source tooling for prediction generation
  – Including data warehousing & machine learning
  – Leverage TheTransitClock (formerly Transitime)
    • [http://thetransitclock.org](http://thetransitclock.org)
Thanks!

Derek Fretheim
derek.fretheim@moovel.com
949.275.6365

Sean J. Barbeau, Ph.D.
barbeau@cutr.usf.edu
813.974.7208

linkedin.com/in/SeanBarbeau
@sjbarbeau

Funding from:
NITC final report - https://nitc.trec.pdx.edu/research/project/1062
WHY SO MANY OPTIONAL FIELDS?
GTFS-rt uses Protocol Buffer (PB) format

- **gtfs-realtime.proto** PB file defines elements to exchange
Protocol Buffers save space

- Compressed binary is around 6 times smaller than plain text
**PB required ≠ transit required**

- v1.0 - *Optional/required* for *Cardinality* field values were copied from `.proto` file

```cpp
31 // Metadata about a feed, included in feed messages.
32 message FeedHeader {
33     // Version of the feed specification.
34     // The current version is 2.0.
35     required string gtfs_realtime_version = 2;
36
37     // Determines whether the current fetch is incremental. Currently,
38     // DIFFERENTIAL mode is unsupported and behavior is unspecified for feeds
39     // that use this mode. There are discussions on the GTFS Realtime mailing
40     // list around fully specifying the behavior of DIFFERENTIAL mode and the
41     // documentation will be updated when those discussions are finalized.
42     enum Incrementality {
43         FULL_DATASET = 0;
44         DIFFERENTIAL = 1;
45     }
46     optional Incrementality incrementality = 2 [default = FULL_DATASET];
47
48     // This timestamp identifies the moment when the content of this feed has been
```