ACT Government
Transport Canberra and City Services

Transport Canberra & City Services

GTFSR - TCCS Implementation Specification

***Version**: 3.1*

Date:	03 July 2018
Version:	3.1
Owner:	Systems Project Manager (Paul Skidmore)
Division:	Transport Canberra (TC)
Review date:	Upon with any agreed changes to TC standards

Contents

1 Overview

1.1 Overview

The purpose of this document is to describe the structure and contents of the Canberra Light Rail realtime data feed supplied by Canberra Metro for consumption and use by Transport Canberra and business units and application developers. The Light Rail realtime data feed is in the form of a General Transit Feed Specification fileset, known as ‘GTFS-R’.

The intended audience of this document is Canberra Metro and application developers. While GTFS-R is well documented, it provides several optional fields and some flexibility in how to populate some fields. This document provides some general information regarding the specific contents and structure of the GTFS-R feed.

A GTFS Realtime feed lets transit agencies provide consumers with realtime information about disruptions to their service (stations closed, lines not operating, important delays, etc.) location of Light Rail, and expected arrival times. GTFS Realtime is a feed specification that allows public transportation agencies to provide realtime updates about their fleet to application developers. It is an extension to GTFS (General Transit Feed Specification), an open data format for public transportation schedules and associated geographic information. GTFS Realtime was designed around ease of implementation, good GTFS interoperability and a focus on passenger information.

The specification currently supports the following types of information:

- Trip updates - delays, cancellations, changed routes
- Service alerts - stop moved, unforeseen events affecting a station, route or the entire network

Vehicle positions - information about the vehicles including location and congestion level.

2 Timetable Data Feed – General Requirements

2.1 Delivery Mode

Feeds are served via HTTP and updated frequently. The file itself is a regular binary file, so any type of webserver can host and serve the file (other transfer protocols might be used as well). Alternatively, web application servers could also be used which as a response to a valid HTTP GET request will return the feed. There are no constraints on how frequently nor on the exact method of how the feed should be updated or retrieved.

Because GTFS Realtime allows you to present the actual status of your fleet, the feed needs to be updated regularly - preferably whenever new data comes in from your Automatic Vehicle Location system.

GTFS-Realtime data shall be published shall be published at the following url: http://files.transport.act.gov.au/feeds/lightrail.pb. These feeds will be registered with Google via Transport Canberra's Google account and Google pull the data from our website location every 30 seconds (Fetch Delivery Method[1]). Third party app developers can also pull the GTFS Realtime feed from Transport Canberra website

2.2 Dependencies

Since the GTFS Realtime is dependent on the static GTFS feed for information, Canberra Metro and Transport Canberra need to work together to implement best practices to get consistent data feeds and adhere to the standard naming conventions of trips, routes and so on

2.3 Data Format

The GTFS Realtime data exchange format is based on Protocol Buffers. Protocol buffers are a language- and platform-neutral mechanism for serializing structured data (think XML, but smaller, faster, and simpler). The data structure is defined in a gtfs-realtime.proto file2, which then is used to generate source code to easily read and write your structured data from and to a variety of data streams, using a variety of languages – e.g. Java, C++ or Python

[1] https://support.google.com/transitpartners/answer/2529132
[2] https://developers.google.com/transit/gtfs-realtime/gtfs-realtime.proto

3 Specifications

3.1 Term definitions

The following protocol buffer data types are used to describe feed elements:

- message: Complex type
- enum: List of fixed values

The following values are used in the Required field:

- Required: This field must be provided by a GTFS-realtime feed producer.
- Conditionally required: This field is required under certain conditions, which are outlined in the field Description. Outside of these conditions, the field is optional.
- Optional: This field is optional and is not required to be implemented by producers. However, if the data is available in the underlying automatic vehicle location systems (e.g., VehiclePosition timestamp) it is recommended that producers provide these optional fields when possible.

3.2 Cardinality

Cardinality represents the number of elements that may be provided for a particular field, with the following values:

- **One** - A single one element may be provided for this field. This maps to the Protocol Buffer *required* and *optional* cardinalities.
- **Many** - Many elements (0, 1, or more) may be provided for this field. This maps to the Protocol Buffer *repeated* cardinality.

Always reference the *Required* and *Description* fields to see when a field is required, conditionally required, or optional.

3.3 Data fields

3.3.4 message FeedMessage

The contents of a feed message. Each message in the stream is obtained as a response to an appropriate HTTP GET request. A realtime feed is always defined with relation to an existing GTFS feed. All the entity ids are resolved with respect to the GTFS feed.

Field Name	Type	Required	Cardinality	Description
header	FeedHeader	Required	One	Metadata about this feed and feed message.
entity	FeedEntity	Required	Many	Contents of the feed. If there is real-time information available for the transit system, this field must be provided. If this field is empty, consumers should assume there is no real-time information available for the system.

3.3.5 message FeedHeader

Metadata about a feed, included in feed messages.

Field Name	Type	Required	Cardinality	Description
gtfs_realtime_version	string	Required	One	Version of the feed specification. The current version is 2.0.
incrementality	Incrementality	Required	One	
timestamp	uint64	Required	One	This timestamp identifies the moment when the content of this feed has been created (in server time). In POSIX time (i.e., number of seconds since January 1st 1970 00:00:00 UTC). To avoid time skew between systems producing and consuming realtime information it is strongly advised to derive timestamp from a time server. It is completely acceptable to use Stratum 3 or even lower strata servers since time differences up to a couple of seconds are tolerable.

3.3.6 enum Incrementality

Determines whether the current fetch is incremental.

FULL**_DATASET**: this feed update will overwrite all preceding realtime information for the feed. Thus this update is expected to provide a full snapshot of all known realtime information.

3.3.7 message FeedEntity

A definition (or update) of an entity in the transit feed. If the entity is not being deleted, exactly one of 'trip_update', 'vehicle' and 'alert' fields should be populated.

Field Name	Type	Required	Cardinality	Description
id	string	Required	One	Feed-unique identifier for this entity. The ids are used only to provide incrementality support. The actual entities referenced by the feed must be specified by explicit selectors (see EntitySelector below for more info).
is_deleted	bool	Optional	One	Whether this entity is to be deleted. Should be provided only for feeds with Incrementality of DIFFERENTIAL - this field should NOT be provided for feeds with Incrementality of FULL_DATASET.
trip_update	TripUpdate	Required	One	Data about the realtime departure delays of a trip. At least one of the fields trip_update, vehicle, or alert must be provided - all these fields cannot be empty.
vehicle	VehiclePosition	Required	One	Data about the realtime position of a vehicle. At least one of the fields trip_update, vehicle, or alert must be provided - all these fields cannot be empty.
alert	Alert	Required	One	Data about the realtime alert. At least one of the fields trip_update, vehicle, or alert must be provided - all these fields cannot be empty.

3.3.8 message TripUpdate

Realtime update on the progress of a vehicle along a trip.

Depending on the value of ScheduleRelationship, a TripUpdate can specify:

- A trip that proceeds along the schedule.
- A trip that proceeds along a route but has no fixed schedule.
- A trip that has been added or removed with regard to schedule.

The updates can be for future, predicted arrival/departure events, or for past events that already occurred. In most cases information about past events is a measured value thus its uncertainty value is recommended to be 0. Although there could be cases when this does not hold so it is allowed to have uncertainty value different from 0 for past events. If an update's uncertainty is not 0, either the update is an approximate prediction for a trip that has not completed or the measurement is not precise or the update was a prediction for the past that has not been verified after the event occurred.

Note that the update can describe a trip that has already completed. To this end, it is enough to provide an update for the last stop of the trip. If the time of arrival at the last stop is in the past, the client will conclude that the whole trip is in the past (it is possible, although inconsequential, to also provide updates for preceding stops). This option is most relevant for a trip that has completed ahead of schedule, but according to the schedule, the trip is still proceeding at the current time. Removing the updates for this trip could make the client assume that the trip is still proceeding. Note that the feed provider is allowed, but not required, to purge past updates - this is one case where this would be practically useful.

Field Name	Type	Required	Cardinality	Description
trip	TripDescriptor	Required	One	The Trip that this message applies to. There can be at most one TripUpdate entity for each actual trip instance. If there is none, that means there is no prediction information available. It does *not* mean that the trip is progressing according to schedule.
vehicle	VehicleDescriptor	Optional	One	Additional information on the vehicle that is serving this trip.
stop_time_update	StopTimeUpdate	Required	Many	Updates to StopTimes for the trip (both future, i.e., predictions, and in some cases, past ones, i.e., those that already happened). The updates must be sorted by stop_sequence, and apply for all the following stops of the trip up to the next specified stop_time_update. At least one stop_time_update must be provided for the trip unless the trip.schedule_relationship is CANCELED - if the trip is canceled, no stop_time_updates need to be provided.
timestamp	uint64	Required	One	Moment at which the vehicle's real-time progress was measured. In POSIX time (i.e., the number of seconds since January 1st 1970 00:00:00 UTC).
delay	int32	Required	One	The current schedule deviation for the trip. Delay should only be specified when the prediction is given relative to some

Field Name	Type	Required	Cardinality	Description
				existing schedule in GTFS. Delay (in seconds) can be positive (meaning that the vehicle is late) or negative (meaning that the vehicle is ahead of schedule). Delay of 0 means that the vehicle is exactly on time. Delay information in StopTimeUpdates take precedent of trip-level delay information, such that trip-level delay is only propagated until the next stop along the trip with a StopTimeUpdate delay value specified. Feed providers are strongly encouraged to provide a TripUpdate.timestamp value indicating when the delay value was last updated, in order to evaluate the freshness of the data. Caution: this field is still experimental, and subject to change. It may be formally adopted in the future.

3.3.9 message StopTimeEvent

Timing information for a single predicted event (either arrival or departure). Timing consists of delay and/or estimated time, and uncertainty.

- delay should be used when the prediction is given relative to some existing schedule in GTFS.
- time should be given whether there is a predicted schedule or not. If both time and delay are specified, time will take precedence (although normally, time, if given for a scheduled trip, should be equal to scheduled time in GTFS + delay).

Uncertainty applies equally to both time and delay. The uncertainty roughly specifies the expected error in true delay (but note, we don't yet define its precise statistical meaning). It's possible for the uncertainty to be 0, for example for trains that are driven under computer timing control.

Field Name	Type	Required	Cardinality	Description
delay	int32	Required	One	Delay (in seconds) can be positive (meaning that the vehicle is late) or negative (meaning that the vehicle is ahead of schedule). Delay of 0 means that the vehicle is exactly on time. Either delay or time must be provided within a StopTimeEvent - both fields cannot be empty.
time	int64	Required	One	Event as absolute time. In POSIX time (i.e., number of seconds since January 1st 1970 00:00:00 UTC). Either delay or time must be provided within a StopTimeEvent - both fields cannot be empty.
uncertainty	int32	Optional	One	If uncertainty is omitted, it is interpreted as unknown. To specify a completely certain prediction, set its uncertainty to 0.

3.3.10 message StopTimeUpdate

Realtime update for arrival and/or departure events for a given stop on a trip. 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.

Field Name	Type	Required	Cardinality	Description
stop_sequence	uint32	Required	One	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.
stop_id	string	Optional	One	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.
arrival	StopTimeEvent	Conditionally required	One	If schedule_relationship is empty or SCHEDULED, either arrival or departure must be provided within a StopTimeUpdate - both fields cannot be empty. arrival and departure may both be empty when schedule_relationship is SKIPPED. If

Field Name	Type	Required	Cardinality	Description
				schedule_relationship is NO_DATA, arrival and departure must be empty.
departure	StopTimeEvent	Conditionally required	One	If schedule_relationship is empty or SCHEDULED, either arrival or departure must be provided within a StopTimeUpdate - both fields cannot be empty. arrival and departure may both be empty when schedule_relationship is SKIPPED. If schedule_relationship is NO_DATA, arrival and departure must be empty.
schedule_relationship	ScheduleRelationship	Optional	One	The default relationship is SCHEDULED.

3.3.11 enum ScheduleRelationship

The relation between this StopTime and the static schedule.

Value	Comment
SCHEDULED	The vehicle is proceeding in accordance with its static schedule of stops, although not necessarily according to the times of the schedule. This is the default behavior. At least one of arrival and departure must be provided. If the schedule for this stop contains both arrival and departure times then so must this update.
SKIPPED	The stop is skipped, i.e., the vehicle will not stop at this stop. Arrival and departure are optional.
NO_DATA	No data is given for this stop. It indicates that there is no realtime information available. When set NO_DATA is propagated through subsequent stops so this is the recommended way of specifying from which stop you do not have realtime information. When NO_DATA is set neither arrival nor departure should be supplied.

3.3.12 message VehiclePosition

Realtime positioning information for a given vehicle.

Field Name	Type	Required	Cardinality	Description
trip	TripDescriptor	Required	One	The Trip that this vehicle is serving. Can be empty or partial if the vehicle can not be identified with a given trip instance.
vehicle	VehicleDescriptor	Required	One	Additional information on the vehicle that is serving this trip. Each entry should have a unique vehicle id.
position	Position	Required	One	Current position of this vehicle.
current_stop_sequence	uint32	Required	One	The stop sequence index of the current stop. The meaning of current_stop_sequence (i.e., the stop that it refers to) is determined by current_status. If current_status is missing IN_TRANSIT_TO is assumed.
stop_id	string	Required	One	Identifies the current stop. The value must be the same as in stops.txt in the corresponding GTFS feed.
current_status	VehicleStopStatus	Required	One	The exact status of the vehicle with respect to the current stop. Ignored if current_stop_sequence is missing.

Field Name	Type	Required	Cardinality	Description
timestamp	uint64	Optional	One	Moment at which the vehicle's position was measured. In POSIX time (i.e., number of seconds since January 1st 1970 00:00:00 UTC).
congestion_level	CongestionLevel	Optional	One	
occupancy_status	OccupancyStatus	Optional	One	The degree of passenger occupancy of the vehicle. Caution: this field is still experimental, and subject to change. It may be formally adopted in the future.

3.3.13 enum VehicleStopStatus

Value	Comment
INCOMING_AT	The vehicle is 5 seconds away from the stop.
STOPPED_AT	The vehicle is standing at the stop.
IN_TRANSIT_TO	The vehicle has departed the previous stop and is in transit.

3.3.14 enum CongestionLevel

Congestion level that is affecting this vehicle.

Value
UNKNOWN_CONGESTION_LEVEL
RUNNING_SMOOTHLY
STOP_AND_GO
CONGESTION
SEVERE_CONGESTION

3.3.15 enum OccupancyStatus

The degree of passenger occupancy for the vehicle.

This field is still **experimental**, and subject to change. It may be formally adopted in the future.

Value	Comment
EMPTY	*The vehicle is considered empty by most measures, and has few or no passengers onboard, but is still accepting passengers.*
MANY_SEATS_AVAILABLE	*The vehicle has a large percentage of seats available. What percentage of free seats out of the total seats available is to be considered large enough to fall into this category is determined at the discretion of the producer.*
FEW_SEATS_AVAILABLE	*The vehicle has a small percentage of seats available. What percentage of free seats out of the total seats available is to be considered small enough to fall into this category is determined at the discretion of the producer.*
STANDING_ROOM_ONLY	*The vehicle can currently accommodate only standing passengers.*
CRUSHED_STANDING_ROOM_ONLY	*The vehicle can currently accommodate only standing passengers and has limited space for them.*
FULL	*The vehicle is considered full by most measures, but may still be allowing passengers to board.*
NOT_ACCEPTING_PASSENGERS	*The vehicle can not accept passengers.*

3.3.16 message Alert

An alert, indicating some sort of incident in the public transit network.

Field Name	Type	Required	Cardinality	Description
active_period	TimeRange	Required	Many	Time when the alert should be shown to the user. If missing, the alert will be shown as long as it appears in the feed. If multiple ranges are given, the alert will be shown during all of them.
informed_entity	EntitySelector	Required	Many	Entities whose users we should notify of this alert. At least one informed_entity must be provided.
cause	Cause	Optional	One	
effect	Effect	Optional	One	
url	TranslatedString	Optional	One	The URL which provides additional information about the alert.
header_text	TranslatedString	Required	One	Header for the alert. This plain-text string will be highlighted, for example in boldface.
description_text	TranslatedString	Required	One	Description for the alert. This plain-text string will be formatted as the body of the alert (or shown on an explicit "expand" request by the user). The information in the description should add to the information of the header.

3.3.17 enum Cause

Cause of this alert.

Value
UNKNOWN_CAUSE
OTHER_CAUSE
TECHNICAL_PROBLEM
STRIKE
DEMONSTRATION
ACCIDENT
HOLIDAY
WEATHER
MAINTENANCE
CONSTRUCTION

Value
POLICE_ACTIVITY
MEDICAL_EMERGENCY

3.3.18 enum Effect

The effect of this problem on the affected entity.

Value
NO_SERVICE
REDUCED_SERVICE
SIGNIFICANT_DELAYS
DETOUR
ADDITIONAL_SERVICE

Value
MODIFIED_SERVICE
OTHER_EFFECT
UNKNOWN_EFFECT
STOP_MOVED

3.3.19 message TimeRange

A time interval. The interval is considered active at time t if it is greater than or equal to the start time and less than the end time.

Field Name	Type	Required	Cardinality	Description
start	uint64	Conditionally required	One	Start time, in POSIX time (i.e., number of seconds since January 1st 1970 00:00:00 UTC). If missing, the interval starts at minus infinity. If a TimeRange is provided, either start or end must be provided - both fields cannot be empty.
end	uint64	Conditionally required	One	End time, in POSIX time (i.e., number of seconds since January 1st 1970 00:00:00 UTC). If missing, the interval ends at plus infinity. If a TimeRange is provided, either start or end must be provided - both fields cannot be empty.

3.3.20 message Position

A geographic position of a vehicle.

Field Name	Type	Required	Cardinality	Description
latitude	float	Required	One	Degrees North, in the WGS-84 coordinate system.
longitude	float	Required	One	Degrees East, in the WGS-84 coordinate system.
bearing	float	Required	One	Bearing, in degrees, clockwise from True North, i.e., 0 is North and 90 is East. This can be the compass bearing, or the direction towards the next stop or intermediate location. This should not be deduced from the sequence of previous positions, which clients can compute from previous data.
odometer	double	Required	One	Odometer value, in meters.
speed	float	Required	One	Momentary speed measured by the vehicle, in meters per second.

3.3.21 message TripDescriptor

A descriptor that identifies an instance of a GTFS trip, or all instances of a trip along a route. To specify a single trip instance, the trip_id (and if necessary, start_time) is set. If route_id is also set, then it should be same as one that the given trip corresponds to. To specify all the trips along a given route, only the route_id should be set. Note that if the trip_id is not known, then station sequence ids in TripUpdate are not sufficient, and stop_ids must be provided as well. In addition, absolute arrival/departure times must be provided.

Field Name	Type	Required	Cardinality	Description
trip_id	string	Conditionally required	One	The trip_id from the GTFS feed that this selector refers to. For non frequency-based trips (trips not defined in GTFS frequencies.txt), this field is enough to uniquely identify the trip. For frequency-based trips defined in GTFS frequencies.txt, trip_id, start_time, and start_date are all required. For scheduled-based trips (trips not defined in GTFS frequencies.txt), trip_id can only be omitted if the trip can be uniquely identified by a combination of route_id, direction_id, start_time, and start_date, and all those fields are provided.
route_id	string	Conditionally required	One	The route_id from the GTFS that this selector refers to. If trip_id is omitted, route_id must be provided.
direction_id	uint32	Conditionally required	One	The direction_id from the GTFS feed trips.txt file, indicating the direction of travel for trips this selector refers to. If trip_id is omitted, direction_id must be provided.

Field Name	Type	Required	Cardinality	Description
				Caution: this field is still experimental, and subject to change. It may be formally adopted in the future.
start_time	string	Conditionally required	One	The initially scheduled start time of this trip instance. When the trip_id corresponds to a non-frequency-based trip, this field should either be omitted or be equal to the value in the GTFS feed. When the trip_id correponds to a frequency-based trip defined in GTFS frequencies.txt, start_time is required and must be specified for trip updates and vehicle positions. If the trip corresponds to exact_times=1 GTFS record, then start_time must be some multiple (including zero) of headway_secs later than frequencies.txt start_time for the corresponding time period. If the trip corresponds to exact_times=0, then its start_time may be arbitrary, and is initially expected to be the first departure of the trip. Once established, the start_time of this frequency-based exact_times=0 trip should be considered immutable, even if the first departure time changes -- that time change may instead be reflected in a StopTimeUpdate. If trip_id is omitted, start_time must be provided. Format and semantics of the field is same as that of GTFS/frequencies.txt/start_time, e.g., 11:15:35 or 25:15:35.

Field Name	Type	Required	Cardinality	Description
start_date	string	Conditionally required	One	The start date of this trip instance in YYYYMMDD format. For scheduled trips (trips not defined in GTFS frequencies.txt), this field must be provided to disambiguate trips that are so late as to collide with a scheduled trip on a next day. For example, for a train that departs 8:00 and 20:00 every day, and is 12 hours late, there would be two distinct trips on the same time. This field can be provided but is not mandatory for schedules in which such collisions are impossible - for example, a service running on hourly schedule where a vehicle that is one hour late is not considered to be related to schedule anymore. This field is required for frequency-based trips defined in GTFS frequencies.txt. If trip_id is omitted, start_date must be provided.
schedule_relationship	ScheduleRelationship	Optional	One	

3.3.22 enum ScheduleRelationship

The relation between this trip and the static schedule. If a trip is done in accordance with temporary schedule, not reflected in GTFS, then it shouldn't be marked as SCHEDULED, but marked as ADDED.

Value	Comment
SCHEDULED	Trip that is running in accordance with its GTFS schedule, or is close enough to the scheduled trip to be associated with it.
ADDED	An extra trip that was added in addition to a running schedule, for example, to replace a broken vehicle or to respond to sudden passenger load.
UNSCHEDULED	A trip that is running with no schedule associated to it - this value is used to identify trips defined in GTFS frequencies.txt with exact_times = 0. It should not be used to describe trips not defined in GTFS frequencies.txt, or trips in GTFS frequencies.txt with exact_times = 1.
CANCELED	A trip that existed in the schedule but was removed.

3.3.23 message VehicleDescriptor

Identification information for the vehicle performing the trip.

Field Name	Type	Required	Cardinality	Description
id	string	Required	One	Internal system identification of the vehicle. Should be unique per vehicle, and is used for tracking the vehicle as it proceeds through the system. This id should not be made visible to the end-user; for that purpose use the labelfield
label	string	Required	One	User visible label, i.e., something that must be shown to the passenger to help identify the correct vehicle.
license_plate	string	Optional	One	The license plate of the vehicle.

3.3.24 message EntitySelector

A selector for an entity in a GTFS feed. The values of the fields should correspond to the appropriate fields in the GTFS feed. At least one specifier must be given. If several are given, then the matching has to apply to all the given specifiers.

Field Name	Type	Required	Cardinality	Description
agency_id	string	Required	One	At least one specifier must be given - all fields in an EntitySelector cannot be empty.

Field Name	Type	Required	Cardinality	Description
route_id	string	Required	One	At least one specifier must be given - all fields in an EntitySelector cannot be empty.
route_type	int32	Required	One	At least one specifier must be given - all fields in an EntitySelector cannot be empty.
trip	TripDescriptor	Required	One	At least one specifier must be given - all fields in an EntitySelector cannot be empty.
stop_id	string	Conditionally required	One	At least one specifier must be given - all fields in an EntitySelector cannot be empty.

3.3.25 message TranslatedString

An internationalized message containing per-language versions of a snippet of text or a URL. One of the strings from a message will be picked up. The resolution proceeds as follows: If the UI language matches the language code of a translation, the first matching translation is picked. If a default UI language (e.g., English) matches the language code of a translation, the first matching translation is picked. If some translation has an unspecified language code, that translation is picked.

Field Name	Type	Required	Cardinality	Description
translation	Translation	Required	Many	At least one translation must be provided.

3.3.26 message Translation

A localized string mapped to a language.

Field Name	Type	Required	Cardinality	Description
text	string	Required	One	A UTF-8 string containing the message.
language	string	Conditionally required	One	BCP-47 language code. Can be omitted if the language is unknown or if no internationalization is done at all for the feed. At most one translation is allowed to have an unspecified language tag - if there is more than one translation, the language must be provided.