

Open Data

DISCLAIMER: The information contained in this dictionary document and in the Open Data Source of this URL might be subject to changes from time to time. The URL provider won't be informing to the possible entities that are using such feeds of any changes, and it's up to them to be properly aligned with the content whenever such content is updated. There won't be other notification apart from the update of the dictionary document itself.

GTFS by fleet

The General Transit Feed Specification (GTFS) is a data specification that allows public transit agencies to publish their transit data in a format that can be consumed by a wide variety of software applications.

The data is offered as a .zip file for each fleet.

GTFS-RT

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). The implemented version is 2.0.

A GTFS Realtime feed is related to an existing static feed, using their entities IDs. Each FeedMessage is the response of an HTTP GET request. This response is returned as a Protobuffer file.

The API allows using an optional argument to get only the data related to the specified operator:

- `api/gtfs-realtime`: returns data for all operators
- `api/gtfs-realtime/1`: returns data for operator 1.

Stop information in .csv format

A csv file is a common data exchange widely supported worldwide. It consist of a text file that uses a delimiter character to separate values.

The stop information can be downloaded as a .csv file. The field delimiter used is a semicolon (;), and the first line defines the headers for each field:

- code: public stop code.
- description: the description of the stop.
- descriptionEL: the description of the stop in Greek (if available).
- descriptionRU: the description of the stop in Russian (if available).
- lat: latitude.
- lon: longitude.

Routes information in .shp format

The shapefile format is a popular geospatial vector data format for geographic information system (GIS) software, allowing the description of vector features: points, lines and polygons.

The routes information is provided using this format.

SIRI-ET

SIRI is a CEN Technical Standard that specifies a European interface standard for exchanging information about the planned, current or projected performance of real-time public transport operations between different computer systems.

Input parameters:

- OperatorRefStructure[] operatorRefField: Fleet name (optional)
- LineDirectionStructure[] linesField
 - LineRefStructure lineRefField: Route to filter the result.
 - DirectionRefStructure directionRefField: Line to filter the result.
- ExtensionsStructure extensionsField: Stop to filter the result.

Output parameters:

- LineRefStructure lineRefField: Public code of route.
- NaturalLanguageStringStructure[] directionNameField
 - string valueField: Line description.
- VehicleJourneyRefStructure vehicleJourneyRefField: Expedition code.
- VehicleRefStructure vehicleRefField: Vehicle code.
- NaturalLanguageStringStructure[] publishedLineNameField
 - string valueField: Route name.
- OperatorRefStructure operatorRefField: Fleet name.
- EstimatedCallStructure[] estimatedCallsField
 - NaturalLanguageStringStructure[] stopPointNameField
 - string valueField: Stop code.
 - DateTime aimedArrivalTimeField: Planned arrival time.
 - DateTime expectedArrivalTimeField: Expected arrival time.
 - DateTime aimedDepartureTimeField: Planned departure time.
 - DateTime expectedDepartureTimeField: Expected departure time.

To access the information, the URL to the public Web Service is provided.

Appendix A. SIRI: Common aspects

Public transport services rely increasingly on information systems to ensure reliable, efficient operation and widely accessible, accurate passenger information. These systems are used for a range of specific purposes: setting schedules and timetables, managing vehicle fleets, issuing tickets and receipts, providing real-time information on service running, and so on.

The Service Interface for Real Time Information (SIRI) specifies a European interface standard for exchanging information about the planned, current or projected performance of real-time public transport operations between different computer systems.

- SIRI comprises a carefully modularized set of discrete functional services for operating public transport information systems. Services cover planned and real time timetable exchange; vehicle activity at stops; vehicle movement; and information to assist in the provision of reliable connections between vehicles.
- SIRI aims to incorporate of the best of various national and proprietary standards from across Europe and deliver these using a modern XML schema and TransModel terminology and modelling concepts.
- All SIRI services are provided over a standardized Communications layer, based on a Web Services Architecture. The Communications layer upholds a consistent approach for all the functional services to Security, Authentication, Version Negotiation, Recovery/Restart, and Access Control/Filtering. To support different operating requirements, two main patterns of interactions are supported: an immediate Request/Response protocol; and an asynchronous Publish/Subscribe protocol. The Publish/Subscribe can be further elaborated with a fetched delivery interaction to optimize the use of bandwidth.

SIRI is extensible and it is expected that additional services will be added over time using the same communications bearer.

SIRI's modularization allows an incremental approach: only the subset of services actually required needs to be implemented for a particular application. The expectation is that users may start with just one or two services and over time increase the number of services and the range for supported options. Similarly Suppliers may extend their support for SIRI in their products incrementally.

SIRI takes a 'joined' up look at all real-time information services, data, data models, transport, and mediation. Considering the whole context is important because for efficiency, real-time services are often only exchanging real-time changes to data, requiring precision about the underlying model assumed by the participants.

A Web Services Discovery process is defined allowing data providers to make the capabilities of their services known to other interested parties.

Appendix B. GTFS spec

This chapter explains the format and structure of the files that comprise a GTFS dataset.

Term Definitions

This section defines terms that are used throughout this document.

- **Dataset** - A complete set of files defined by this specification reference. Altering the dataset creates a new version of the dataset. Datasets should be published at a public, permanent URL, including the zip file name. (e.g., <https://www.agency.org/gtfs/gtfs.zip>).
- **Record** - A basic data structure comprised of a number of different field values describing a single entity (e.g. transit agency, stop, route, etc.). Represented, in a table, as a row.
- **Field** - A property of an object or entity. Represented, in a table, as a column.
- **Field Value** - An individual entry in a field. Represented, in a table, as a single cell.
- **Required** - The field must be included in the dataset, and a value must be provided in that field for each record. Some required fields permit an empty string as a value (denoted in this specification as empty). To enter an empty string, just omit any text between the commas for that field.
- **Service day** - A service day is a time period used to indicate route scheduling. The exact definition of service day varies from agency to agency but service days often do not correspond with calendar days. A service day may exceed 24:00:00 if service begins on one day and ends on a following day. For example, service that runs from 08:00:00 on Friday to 02:00:00 on Saturday, could be denoted as running from 08:00:00 to 26:00:00 on a single service day.

Field Types

- **Color** - A color encoded as a six-digit hexadecimal number. Refer to <https://htmlcolorcodes.com> to generate a valid value (the leading "#" is not included).
 - Example: FFFFFFF for white, 000000 for black or 0039A6 for the A,C,E lines in NYMTA.
- **Date** - Service day in the YYYYMMDD format. Since time within a service day can be above 24:00:00, a service day often contains information for the subsequent day(s).
 - Example: 20180913 for September 13th, 2018.
- **Email** - An email address.
 - Example: `example@example.com`
- **Enum** - An option from a set of predefined constants defined in the "Description" column.
 - Example: The `route_type` field contains a 0 for tram, a 1 for subway...
- **ID** - An ID field value is an internal ID, not intended to be shown to riders, and is a sequence of any UTF-8 characters. Using only printable ASCII characters is recommended. IDs defined in one .txt file are often referenced in another .txt file.
 - Example: The `stop_id` field in `stops.txt` is a ID. The `stop_id` field in `stop_times.txt` is an ID referencing `stops.stop_id`.

- Language Code - An IETF BCP 47 language code. For an introduction to IETF BCP 47, refer to <http://www.rfc-editor.org/rfc/bcp/bcp47.txt> and <http://www.w3.org/International/articles/language-tags/>.
 - Example: en for English, en-US for American English or de for German.
- Latitude - WGS84 latitude in decimal degrees. The value must be greater than or equal to -90.0 and less than or equal to 90.0.
 - Example: 41.890169 for the Colosseum in Rome.
- Longitude - WGS84 longitude in decimal degrees. The value must be greater than or equal to -180.0 and less than or equal to 180.0.
 - Example: 12.492269 for the Colosseum in Rome.
- Non-negative Integer - A integer greater than or equal to 0.
- Phone number - A phone number.
- Time - Time in the HH:MM:SS format (H:MM:SS is also accepted). The time is measured from "noon minus 12h" of the service day (effectively midnight except for days on which daylight savings time changes occur). For times occurring after midnight, enter the time as a value greater than 24:00:00 in HH:MM:SS local time for the day on which the trip schedule begins.
 - Example: 14:30:00 for 2:30PM or 25:35:00 for 1:35AM on the next day.
- Text - A string of UTF-8 characters, which is aimed to be displayed and which must therefore be human readable.
- Timezone - TZ timezone from the <https://www.iana.org/time-zones>. Timezone names never contain the space character but may contain an underscore. Refer to http://en.wikipedia.org/wiki/List_of_tz_zones for a list of valid values.
 - Example: Asia/Tokyo, America/Los_Angeles or Africa/Cairo.
- URL - A fully qualified URL that includes <http://> or <https://>, and any special characters in the URL must be correctly escaped. See the following http://www.w3.org/Addressing/URL/4_URI_Recommentations.html for a description of how to create fully qualified URL values.

File Requirements

The following requirements apply to the format and contents of the dataset files:

- All files must be saved as comma-delimited text.
- The first line of each file must contain field names. Each subsection of the Field Definitions section corresponds to one of the files in a GTFS dataset and lists the field names that may be used in that file.
- All field names are case-sensitive.
- Field values may not contain tabs, carriage returns or new lines.
- Field values that contain quotation marks or commas must be enclosed within quotation marks. In addition, each quotation mark in the field value must be preceded with a quotation mark. This is consistent with the manner in which Microsoft Excel outputs comma-delimited (CSV) files. For more information on the CSV file format, see <http://tools.ietf.org/html/rfc4180>. The following example demonstrates how a field value would appear in a comma-delimited file:
 - Original field value: Contains "quotes", commas and text

- Field value in CSV file: "Contains ""quotes"", commas and text"
- Field values must not contain HTML tags, comments or escape sequences.
- Remove any extra spaces between fields or field names. Many parsers consider the spaces to be part of the value, which may cause errors.
- Each line must end with a CRLF or LF linebreak character.
- Files should be encoded in UTF-8 to support all Unicode characters. Files that include the Unicode byte-order mark (BOM) character are acceptable. See http://unicode.org/faq/utf_bom.html#BOM for more information on the BOM character and UTF-8.
- All dataset files must be zipped together.

Field Definitions

agency.txt

Defines Transit agencies with service represented in this dataset.

Field Name	Type	Description
agency_id	ID	Identifies a transit brand which is often synonymous with a transit agency. Note that in some cases, such as when a single agency operates multiple separate services, agencies and brands are distinct. This document uses the term "agency" in place of "brand". A dataset may contain data from multiple agencies.
agency_name	Text	Full name of the transit agency.
agency_url	URL	URL of the transit agency.
agency_timezone	Timezone	Timezone where the transit agency is located. If multiple agencies are specified in the dataset, each must have the same agency_timezone.
agency_lang	Language code	Primary language used by this transit agency. This field helps GTFS consumers choose capitalization rules and other language-specific settings for the dataset.

stops.txt

Stops where vehicles pick up or drop off riders. Also defines stations and station entrances.

Field Name	Type	Description
stop_id	ID	Identifies a stop, station, or station entrance. The term "station entrance" refers to both station entrances and station exits. Stops, stations or station entrances are collectively referred to as locations. Multiple routes may use the same stop.
stop_code	Text	Short text or a number that identifies the location for riders.
stop_name	Text	Name of the location. Use a name that people will understand in the local and tourist vernacular.
stop_desc	Text	This field is used to specify the short name of the stop.
stop_lat	Latitude	Latitude of the location. The coordinates must be the ones of the bus pole — if exists — and otherwise of where the travelers are boarding the vehicle (on the sidewalk or the platform, and not on the roadway or the track where the vehicle stops).

Field Name	Type	Description
stop_lon	Longitude	Longitude of the location. The coordinates must be the ones of the bus pole — if exists — and otherwise of where the travelers are boarding the vehicle (on the sidewalk or the platform, and not on the roadway or the track where the vehicle stops).

routes.txt

Transit routes. A route is a group of trips that are displayed to riders as a single service.

Field Name	Type	Description
route_id	ID	Identifies a route.
agency_id	ID referencing agency.agency_id	Agency for the specified route. This field is required when the dataset provides data for routes from more than one agency in agency.txt, otherwise it is optional.
route_short_name	Text	Short name of a route. This will often be a short, abstract identifier like "32", "100X", or "Green" that riders use to identify a route, but which doesn't give any indication of what places the route serves. Either route_short_name or route_long_name must be specified, or potentially both if appropriate.
route_long_name	Text	Full name of a route. This name is generally more descriptive than the route_short_name and often includes the route's destination or stop. Either route_short_name or route_long_name must be specified, or potentially both if appropriate.
route_desc	Text	Description of a route that provides useful, quality information.
route_type	Enum	Indicates the type of transportation used on a route. Valid options are: 0 - Tram, Streetcar, Light rail. Any light rail or street level system within a metropolitan area. 1 - Subway, Metro. Any underground rail system within a metropolitan area. 2 - Rail. Used for intercity or long-distance travel. 3 - Bus. Used for short- and long-distance bus routes. 4 - Ferry. Used for short- and long-distance boat service. 5 - Cable car. Used for street-level cable cars where the cable runs beneath the car. 6 - Gondola, Suspended cable car. Typically used for aerial cable cars where the car is suspended from the cable. 7 - Funicular. Any rail system designed for steep inclines.
route_color	Color	Route color designation that matches public facing material. Defaults to white (FFFFFF) when omitted or left empty. The color difference between route_color and route_text_color should provide sufficient contrast when viewed on a black and white screen.
route_text_color	Color	Legible color to use for text drawn against a background of route_color. Defaults to black (000000) when omitted or left empty. The color difference between route_color and route_text_color should provide sufficient contrast when viewed on a black and white screen.

trips.txt

Trips for each route. A trip is a sequence of two or more stops that occur during a specific time period.

Field Name	Type	Description
route_id	ID referencing routes.route_id	Identifies a route.
service_id	ID referencing calendar.service_id or calendar_dates.service_id	Identifies a set of dates when service is available for one or more routes.
trip_id	ID	Identifies a trip.
trip_headsign	Text	Text that appears on signage identifying the trip's destination to riders.
shape_id	ID referencing shapes.shape_id	Identifies a geospatial shape describing the vehicle travel path for a trip.
direction_id	Enum	Indicates the direction of travel for a trip. This field is not used in routing; it provides a way to separate trips by direction when publishing time tables. Valid options are: 0 - Travel in one direction (e.g. outbound travel). 1 - Travel in the opposite direction (e.g. inbound travel).

stop_times.txt

Times that a vehicle arrives at and departs from stops for each trip.

Field Name	Type	Description
trip_id	ID referencing trips.trip_id	Identifies a trip.
arrival_time	Time	Arrival time at a specific stop for a specific trip on a route. If there are not separate times for arrival and departure at a stop, enter the same value for arrival_time and departure_time. For times occurring after midnight on the service day, enter the time as a value greater than 24:00:00 in HH:MM:SS local time for the day on which the trip schedule begins.
departure_time	Time	Departure time from a specific stop for a specific trip on a route. For times occurring after midnight on the service day, enter the time as a value greater than 24:00:00 in HH:MM:SS local time for the day on which the trip schedule begins. If there are not separate times for arrival and departure at a stop, enter the same value for arrival_time and departure_time. See the arrival_time description for more details about using timepoints correctly.
stop_id	ID referencing stops.stop_id	Identifies the serviced stop. All stops serviced during a trip must have a record in stop_times.txt. Referenced locations must be stops, not stations or station entrances. A stop may be serviced multiple times in the same trip, and multiple trips and routes may service the same stop.
stop_sequence	Non-negative integer	Order of stops for a particular trip. The values must increase along the trip but do not need to be consecutive. Example: The first location on the trip could have a stop_sequence=1, the second location on the trip could have a stop_sequence=23, the third location could have a stop_sequence=40, and so on.

Field Name	Type	Description
pickup_type	Enum	Indicates pickup method. Valid options are: 0 or empty - Regularly scheduled pickup. 1 - No pickup available. 2 - Must phone agency to arrange pickup. 3 - Must coordinate with driver to arrange pickup.
drop_off_type	Enum	Indicates drop off method. Valid options are: 0 or empty - Regularly scheduled drop off. 1 - No drop off available. 2 - Must phone agency to arrange drop off. 3 - Must coordinate with driver to arrange drop off.

calendar_dates.txt

Contains all dates of service. The calendar_dates.txt table can explicitly activate or disable service by date by specifying each date of service. This allows for considerable service variation and accommodates service without normal weekly schedules.

Field Name	Type	Description
service_id	ID	Identifies a set of dates when a service exception occurs for one or more routes.
date	Date	Date when service exception occurs.
exception_type	Enum	Indicates whether service is available on the date specified in the date field. Valid options are: 1 - Service has been added for the specified date. 2 - Service has been removed for the specified date. Example: Suppose a route has one set of trips available on holidays and another set of trips available on all other days. One service_id could correspond to the regular service schedule and another service_id could correspond to the holiday schedule. For a particular holiday, the calendar_dates.txt file could be used to add the holiday to the holiday service_id and to remove the holiday from the regular service_id schedule.

shapes.txt

Rules for mapping vehicle travel paths, sometimes referred to as route alignments.

Shapes describe the path that a vehicle travels along a route alignment, and are defined in the file shapes.txt. Shapes are associated with Trips, and consist of a sequence of points through which the vehicle passes in order. Shapes do not need to intercept the location of Stops exactly, but all Stops on a trip should lie within a small distance of the shape for that trip, i.e. close to straight line segments connecting the shape points.

Field Name	Type	Description
shape_id	ID	Identifies a shape.
shape_pt_lat	Latitude	Latitude of a shape point. Each record in shapes.txt represents a shape point used to define the shape.
shape_pt_lon	Longitude	Longitude of a shape point.

Field Name	Type	Description
shape_pt_sequence	Non-negative integer	Sequence in which the shape points connect to form the shape. Values must increase along the trip but do not need to be consecutive. Example: If the shape "A_shp" has three points in its definition, the shapes.txt file might contain these records to define the shape: shape_id,shape_pt_lat,shape_pt_lon,shape_pt_sequence A_shp,37.61956,-122.48161,0 A_shp,37.64430,-122.41070,6 A_shp,37.65863,-122.30839,11

feed_info.txt

This file contains information about the dataset itself, rather than the services the dataset describes. In some cases, the publisher of the dataset differs from the agencies.

Field Name	Type	Description
feed_publisher_name	Text	Full name of the organization that publishes the dataset.
feed_publisher_url	URL	URL of the dataset publishing organization's website.
feed_lang	Language code	Default language for the text in this dataset.

translations.txt

Contains translations for different entities.

Field Name	Type	Description
table_name	Enum	Defines the dataset table that contains the field to be translated: stops routes trips
field_name	Text	Provides the name of the field to be translated. stops: stop_name, stop_desc routes: route_long_name trips: trip_headsign
language	Language code	Provides the language of translation.
translation	Text	Provides the translated value for the specified field_name.
record_id	ID	Defines the record that corresponds to the field to be translated: stops: stop_id routes: route_id trips: trip_id

Appendix B. GTFS-RT spec

This chapter explains the format and structure of the files that comprise a GTFS-RT feed.

A GTFS Realtime feed (FeedMessage) consists of two basic elements, a header (FeedHeader) containing the feed metadata, and a body (FeedEntity) which includes the feed contents.

FeedMessage

The contents of a feed message.

Field name	GTFS type	Cardinality	Description
header	FeedHeader	One	Metadata about this feed and feed message.
entity	FeedEntity	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.

FeedHeader

Metadata about a feed, included in feed messages.

FeedHeader

Field name	GTFS type	Cardinality	Description
gtfs_realtime_version	string	One	Version of the feed specification. The current version is 2.0.
incrementality	Incrementality (enum)	One	Only FULL_DATASET is included.
timestamp	uint64	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).

FeedEntity

A definition (or update) of an entity in the transit feed. Each entity must include one and only one of the fields “trip_update”, “vehicle”.

FeedEntity

Field name	GTFS type	Cardinality	Description
id	string	One	Feed-unique identifier for this entity. The ids are used only to provide incrementality support.
trip_update	TripUpdate	One or zero	Data about the realtime departure delays of a trip.
vehicle	VehiclePosition	One or zero	Data about the realtime position of a vehicle.

TripUpdate

Realtime update on the progress of a vehicle along a trip. The updates can be for future, predicted arrival/departure events, or for past events that already occurred.

TripUpdate

Field name	GTFS type	Cardinality	Description
trip	TripDescriptor	One	The expedition that this message applies to. There can be at most one TripUpdate entity for each actual trip instance.
vehicle	VehicleDescriptor	One	Additional information on the vehicle that is serving this trip.
stop_time_update	StopTimeUpdate	Many	Updates to StopTimes for the trip (both future, i.e., predictions, and in some cases, past ones, i.e., those that already happened).
timestamp	uint64	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).

TripDescriptor

Field name	GTFS type	Cardinality	Description
trip_id	string	One or zero	The trip_id from the GTFS feed that this selector refers to.
route_id	string	One	The route_id from the GTFS feed that this selector refers to.
direction_id	uint3	One	The direction_id from the GTFS feed trips.txt file, which indicates the direction of travel (inbound/outbound).
start_time	string	One	The initially scheduled start time of this trip instance (hh:mm:ss).
start_date	string	One	The start date of this trip instance in YYYYMMDD format.
schedule_relationship	ScheduleRelationship (enum)	One	The relation between this trip and the static schedule: SCHEDULED if the trip exists on the static feed, ADDED if an extra trip was added.

VehicleDescriptor

Field name	GTFS type	Cardinality	Description
id	string	One	Internal system identification of the vehicle. This id should not be made visible to the end-user
label	string	One or zero	User visible label, i.e., something that must be shown to the passenger to help identify the correct vehicle.
license_plate	string	One or zero	The license plate of the vehicle.

StopTimeUpdate

Field name	GTFS type	Cardinality	Description
stop_sequence	uint32	One	Must be the same as in stop_times.txt in the corresponding GTFS feed.
stop_id	string	One	Must be the same as in stops.txt in the corresponding GTFS feed.
arrival	StopTimeEvent	One or zero	Timing information for an arrival event.
departure	StopTimeEvent	One or zero	Timing information for a departure event.

StopTimeEvent

Field name	GTFS type	Cardinality	Description
time	int64	One	Event as absolute time. In POSIX time (i.e., number of seconds since January 1st 1970 00:00:00 UTC).
delay	int32	One	The delay field (in seconds) can be positive (meaning that the vehicle is late) or negative (meaning that the vehicle is ahead of schedule). A delay of 0 means that the vehicle is exactly on time.

VehiclePosition

Realtime positioning information for a given vehicle.

VehiclePosition

Field name	GTFS type	Cardinality	Description
trip	TripDescriptor	One or zero	The Expedition that this vehicle is serving. Can be empty or partial if the vehicle cannot be identified with a given trip instance.
vehicle	VehicleDescriptor	One	Additional information on the vehicle that is serving this trip.
position	Position	One	Current position of this vehicle.
stop_id	string	One or zero	Identifies the current stop. The value must be the same as in stops.txt in the corresponding GTFS feed.
timestamp	uint64	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).

Position

Field name	GTFS type	Cardinality	Description
latitude	float	One	Degrees North, in the WGS-84 coordinate system.
longitude	float	One	Degrees East, in the WGS-84 coordinate system.