



Mobility Data Specification (MDS) and General Transit Feed Specification (GTFS)

Updated: May 4, 2026

In FY 2026-2027, SFMTA is going to replace our existing custom Commuter Shuttle API by moving entirely to the Mobility Data Specification (MDS) industry standard. The SFMTA collects data on shuttle vehicles, trips, and provides data to Permittees through data feeds built according to the Mobility Data Specification (MDS). The Open Mobility Foundation (OMF) manages MDS, and SFMTA is a founding and active member of OMF. The SFMTA encourages Permittees to also contribute to future changes to MDS through OMF's public process.

Draft

Permittees are required to provide SFMTA with accurate and authenticated data about its entire permitted fleet through documented Application Programming Interfaces (APIs) built to the latest Mobility Data Specification (MDS) version 2.0.1 available at: <https://github.com/openmobilityfoundation/mobility-data-specification>

In addition to MDS, the SFMTA will require permittees to support the [General Transit Feed Specification \(GTFS\)](#). We are aware that all shuttle operators and their customers already use GTFS for scheduling and routing. Since MDS does not contain a route identifier field and GTFS does, the SFMTA will access the GTFS for commuter shuttle route planning so that we can capture the route information in a more efficient way to save time for all parties.

Permittee must provide data for all devices operating within the City and County of San Francisco. Permittees shall retain data and keep it accessible via all required endpoints for at least two years after it is generated.

The MDS standard establishes a broad array of data elements. The SFMTA only requires a subset of the MDS data elements, as described below.

SFMTA supports pulling and receiving data from providers so we can pull historical data if there is missing data. The SFMTA requires Permittees to provide data through the following MDS APIs and endpoints:

Data Provided by Permittee (MDS Provider API and GTFS) and SFMTA (MDS Agency API)

MDS Provider API:

- Events endpoint (MDS fields used for SFMTA Commuter Shuttle Program)

Field	Type	Required/Optional	Field Description
device_id	UUID	Required	A unique device ID in UUID format

provider_id	UUID	Required	A UUID for the Provider, unique within MDS. See MDS provider list.
data_provider_id	UUID	Required	If different than provider_id, a UUID for the data solution provider managing the data feed in this endpoint. See MDS provider list which includes both service operators and data solution providers.
event_id	UUID	Required	A unique event ID
vehicle_state	Enum	Required	See vehicle state table
event_types	Enum[]	Required	Vehicle event types for state change, with allowable values determined by vehicle state
timestamp	Timestamp	Required	Date/time that event occurred at. See Event Times
publication_time	Timestamp	Optional	Date/time that event became available through the status changes endpoint
location	GPS	Required	See also Telemetry.
event_geographies	UUID[]	Optional	Beta feature: Yes (as of 2.0.0). Array of Geography UUIDs consisting of every Geography that contains the location of the status change. See Geography Driven Events. Required if location is not present.
trip_ids	UUID[]	Required if Applicable	Trip UUIDs (foreign key to /trips endpoint), required if event_types contains trip_start, trip_end, trip_cancel, trip_enter_jurisdiction, or trip_leave_jurisdiction
stop_id	UUID	Required	Stop that the vehicle is currently located at. See Stops

- Telemetry endpoint (MDS fields used for SFMTA Commuter Shuttle Program)

Field	Type	Required/Optional	Field Description
device_id	UUID	Required	A unique device ID in UUID format
provider_id	UUID	Required	https://github.com/openmobilityfoundation/mobility-data-specification/blob/dev/providers.csv
data_provider_id	UUID	Required by SFMTA	If different than provider_id, a UUID for the data solution provider managing the data feed in this endpoint. See MDS provider list which includes both service operators and data solution providers.
telemetry_id	UUID	Required	ID used for uniquely-identifying a Telemetry entry
timestamp	Timestamp	Required	Date/time that event occurred. Based on GPS or GNSS clock

trip_ids	UUID[]	Required	If telemetry occurred during a trip, the ID of the trip(s). If not in a trip, null
journey_id	UUID	Required	If telemetry occurred during a trip and journeys are used for the mode, the ID of the journey. If not in a trip, null
stop_id	UUID	Required if Applicable	Stop that the vehicle is currently located at. See Stops
location	GPS	Required	Telemetry position data
lat	Double	Required	Latitude of the location (from GPS)
lng	Double	Required	Longitude of the location (from GPS)
location_type	Enum	Required if Known	If detectable and known, what type of location the device is on or in. One of street, sidewalk, crosswalk, garage, bike_lane.

- Trips endpoint (MDS fields used for SFMTA Commuter Shuttle Program)

Field	Type	Required/Optional	Field Description
provider_id	UUID	Required	A UUID for the Provider, unique within MDS. See MDS provider list.
data_provider_id	UUID	Required	If different than provider_id, a UUID for the data solution provider managing this data endpoint. See MDS provider list which includes both service operators and data solution providers.
device_id	UUID	Required	A unique device ID in UUID format. Cross reference with /vehicles for more device details.
journey_id	UUID	Optional	A unique <u>journey ID</u> for associating collections of trips for its <u>mode</u>
journey_attributes	Map	Optional	Mode Specific. <u>Journey attributes</u> given as unordered key-value pairs
trip_id	UUID	Required	A unique ID for each trip
trip_type	Enum	Optional	Mode Specific. The <u>trip type</u> describing the purpose of a trip segment
trip_attributes	Map	Required by SFMTA for the passenger_count and wheelchair_transported	Mode Specific. <u>Trip attributes</u> given as unordered key-value pairs (passenger_count (integer, required): unique count of passengers transported during trip duration and wheelchair_transported (boolean,

			Optional) - was a wheelchair transported as part of this trip?). <u>SFMTA required this field with value</u>
start_time	Timestamp	Required	Start of the passenger/driver trip
end_time	Timestamp	Required	End of the passenger/driver trip
start_location	GPS	Required	Location of the start of the trip.
end_location	GPS	Required	Location of the end of the trip.
duration	Integer	Required	Time, in Seconds
distance	Integer	Required	Trip Distance, in Meters
publication_time	Timestamp	Optional	Date/time that trip became available through the trips endpoint
gtfs_trip_id	String	Required	A unique trip ID for the associated scheduled GTFS route-trip. Matches GTFS trip_id in the trips.txt and other files.
gtfs_api_url	URL	Required	Full URL to the location where the associated GTFS dataset zip files are located.

MDS Agency API:

- Events
- Telemetry
- Trips
- Stops
- Vehicle-Register
- Vehicle-Update
- Vehicle-List
- Geographies

GTFS:

- trips.txt
- stops.txt
- stop_times.txt
- shapes.txt
- routes.txt
- feed_info.txt
- calendar_dates.txt
- calendar.txt
- agency.txt

MDS Service Level Standards

SFMTA requires the following service level standards for the availability of and transmission of MDS data to ensure timely reporting:

- **Latency:** data will be pushed via the Agency API, as defined in MDS, and available in the Provider API, as defined in MDS, (as applicable) no more than one hour after events have occurred.
- **Availability:** the API endpoints will be available 99.5% of the time over the course of a year.
- **Response Time:** 85% of Transmission Control Protocol (TCP) API replies within 1.5 seconds, and 99.5% of TCP replies within 4 seconds of receiving a request over the course of a month.

Draft

Data Provided by SFMTA

The SFMTA provides data in a standard format to help Permittees conform to SFMTA's requirements. The SFMTA requires that Permittees pull data from these APIs.

Geography API (includes shuttle designated stops, arterials streets etc.):

- Arterial streets
<https://services.sfmta.com/arcgis/rest/services/ROW/rightofway/MapServer/4>
- Slow streets
<https://services.sfmta.com/arcgis/rest/services/Traffic/traffic/MapServer/13>
- Restricted shuttle streets
<https://services.sfmta.com/arcgis/rest/services/Traffic/traffic/FeatureServer/11>
- Shuttle stops
https://services.sfmta.com/shuttle/2_0/stops/
<https://services.sfmta.com/arcgis/rest/services/Traffic/traffic/MapServer/2>

Policy API – Requirements endpoint (includes these data specification requirements in a machine-readable form) Policy

- The latest version of SFMTA's requirements file can be found at:
<https://github.com/openmobilityfoundation/agency-program-requirements/tree/main/requirement>

Vehicle States Events

This is the list of vehicle_state and event_type pairings that constitute the valid transitions of the vehicle state machine for **commuter** shuttle vehicles with multiple stops to pickup and dropoff passengers.

<https://github.com/openmobilityfoundation/mobility-data-specification/blob/main/modes/passenger-services.md#vehicle-states-events>

From vehicle_state	To vehicle_state	trip_state	event_type	stop_id	Description
stopped	on_trip	on_trip	trip_start	N/A	Start a trip
on_trip	stopped	stopped	trip_stop	108189CB-179E-4AAC-A5A7-02BE6D8EC27A	The vehicle has stopped at shuttle stop while on a trip to pick up or drop off passenger
stopped	on_trip	on_trip	trip_resume	N/A	Resume a trip that was previously stopped (e.g. picking up or drop off passenger)
on_trip	stopped	stopped	trip_stop	923CD797-310A-44D8-B2B6-3E8137A142AC	The vehicle has stopped at shuttle stop while on a trip to pick up or drop off passenger
stopped	on_trip	on_trip	trip_resume	N/A	Resume a trip that was previously stopped (e.g. picking up or drop off passenger)
on_trip	stopped	stopped	trip_stop	A6630DB3-50B0-4B0A-B147-E4D6B4CB09A4	The vehicle has stopped at shuttle stop while on a trip to pick up or drop off passenger
stopped	on_trip	on_trip	trip_resume	N/A	Resume a trip that was previously stopped (e.g. picking up or drop off passenger)
stopped	available	N/A	trip_end	N/A	The shuttle trip has been successfully completed

Authentication Instructions for sending MDS to testing and production server

- Credentials info for each shuttle operator is created and sent to shuttle operators and tech providers.

Existing API fields equivalent mapping to MDS 2.0

Stop

Field in MDS	Type in MDS	Required/Optional in MDS	Field Description in MDS	Field in existing API	Description in existing API
stop_id	UUID	Required for commuter shuttle.	Stop that the vehicle is currently located at. See Stops	StopId	The stop id, normally a Muni stop identifier as used by the SFMTA
Short_name	string	Required for commuter shuttle. This will be same existing stop# eg. 13735			
name	string	Required	Name of the stop Eg. Bryant St&18th St SE corner, near-side stop	StopDescription	The description of the stop
lat	Double	Required	Latitude of the stop location	StopLocationLatitude	The geo latitude of the stop id location in WGS84 latitude
lng	Double	Required	Longitude of the stop location	StopLocationLongitude	The geo longitude of the stop id location in WGS84 longitude

Telemetry

Field in MDS	Type in MDS	Required/Optional in MDS	Field Description in MDS	Field in existing API	Description in existing API
data_provider_id	UUID	Optional	If different than provider_id, a UUID for the data solution provider managing this data endpoint. See MDS provider list which includes both service operators and data solution providers.	TechProviderId	Identification number of technology provider as assigned by SFMTA
provider_id	UUID	Required	A UUID for the Provider, unique within MDS. See MDS provider list.	ShuttleCompanyId	Identification number of shuttle provider as assigned by SFMTA.
placard_number	string	Optional	the registered shuttle permit stickers with unique number in 00-000 format. The first 2 digit identify shuttle company. The last four issue in sequence number eg. 05-1255	VehiclePlacardNum	The placard number issued to each vehicle. Must be unique for each vehicle.
vehicle_id	String	Required	A unique vehicle identifier (visible code, license plate,	LicensePlateNum	License plate number issued to each vehicle.

			etc), visible on the vehicle itself		
stop_id	UUID	Required for commuter shuttle.	Stop that the vehicle is currently located at. See Stops	StopId	The stop id used by SFMTA. See AllowedStops API for list of allowed stop ids. For temporary stops or stops made outside of the shuttle network use ID 9999
Short_name	string	Required for commuter shuttle. This will be same existing stop# eg. 13735			
vehicle_state	Enum	Required	See vehicle state table	VehicleStatus	The vehicle status
lat	Double	Required	Latitude of the location	LocationLatitude	The geo latitude of the vehicle.
lng	Double	Required	Longitude of the location	LocationLongitude	The geo longitude of the vehicle.
timestamp	Timestamp	Required	Date/time that event occurred. Based on GPS or GNSS clock	TimeStampLocal	Date and time vehicle reported this location
route_id	UUID	Required	No route_id in MDS. Provider needs to provide GTFS URL and GTFS trip_id in trip endpoint to pull the route_id from GTFS	RouteIdentifier	Unique identifier for the route. The identifier should start with the provider id, next the same name as provided in the application, followed by the time period. For example, 00_Mission_AM
gtfs_trip_id	UUID				
gtfs_api_url	string				
event_type	string	Required	Examples: Trip enters a jurisdiction (San Francisco) = trip_enter_jurisdiction Trip enters a jurisdiction (San Francisco) = trip_leave_jurisdiction	RouteType	Inter or intracity route 1 – Intercity (within SF) 2 – Intracity (between cities)
trip_id	UUID	Required	A unique ID for each trip	RunID	Unique number for each run
passenger_count	integer	Required	unique count of passengers transported during trip duration	RunOccupancy	Current number of passengers onboard the vehicle during the trip.

General Transit Feed Specification (GTFS)

Per the [Commuter Shuttle Permit Terms and Conditions](#), SFMTA requires route data (see VIII. Operating Guidelines: Operators of Shuttle Buses authorized under this permit shall obey the following Operating Guidelines; IX. Data Reporting Requirements) we need to know the route planning for each shuttle service to be dispatched in accordance with SFMTA guidelines.

We currently use two elements to verify route planning. The first is the route identifier field in the existing API. The second is that each shuttle operator draws the routes on the map for SFMTA to review and approve the route planning. However, the [route maps](#) have not been updated since COVID and the route may not be accurate.

SFMTA would like to access the GTFS for commuter shuttle route planning so that we can more efficiently capture the route_id, service_id and other route information to save time for all parties.

SFMTA plans to use the shape.txt in GTFS to replace the existing manually drawn route map. This could allow the routes on the map to be updated more effectively and accurately. We share the commuter shuttle route map with other agencies and departments when there are construction, road changes, and special events that may affect planned commuter shuttle routes so that they can provide advance notice to the affected commuter shuttle routes by contacting the shuttle operators. The SFMTA only shares route data for the Commuter Shuttle Program to better communicate events and construction that may affect a specific route.

The SFMTA is not adding anything new to the Community Shuttle Program to access the existing GTFS data already used by all shuttle operators. We are trying to use technology to help manage the programme in a more streamlined way in terms of data reporting, saving time for all stakeholders where we can, and connecting to MDS 2.0 to improve data reporting. Where credentials are required to access GTFS, SFMTA will use them internally to access the data linked to the MDS database.