

SFMTA Taxi API 3.2.2 Specifications

November 1, 2023

The following document outlines the API specifications to be used by Dispatch Companies and Payment System Providers to send Taxi Electronic Trip Data and Telemetry Data to the SFMTA in compliance with the data requirements specified both in the Transportation Code and as set forth by the Director of Transportation in this document.

Applicable articles from Section 1114 of the Transportation Code:

(1) **Electronic Trip Data.** Each Dispatch Service Permit Holder shall implement a system or enhance an existing system to generate Electronic Trip Data in a format approved by the SFMTA for all affiliated vehicles. Such system must, at a minimum, archive all taxi trip data for at least five years, produce data that can generate reports using commonly available database and spreadsheet software, and record the following information:

- (A) Driver's identification established by authentication through Driver's license swipe or other secure system;
- (B) Date of shift;
- (C) Vehicle number, vehicle license number, and vehicle status (available or hired);
- (D) Medallion number (manually entered);
- (E) GPS-generated origin, incremental, destination coordinates of each trip;
- (F) The fare for each trip including applicable fees charged and method of payment, which may include, but shall not be limited to, the following: cash, credit card, debit card, voucher, and mobile payment;
- (G) The distance traveled for each trip; and
- (H) The local time of hire and discharge for each trip.

(2) **Integration with Electronic Taxi Access System.**

(A) Each Dispatch Service Permit Holder shall implement a system or extend an existing system to integrate and exchange Electronic Trip Data with the Electronic Taxi Access System.

(B) Each Dispatch Service Permit Holder shall use systems to share Electronic Trip Data in real-time. As Drivers operate the vehicle, systems used by Dispatch Service Permit Holders shall exchange taxi trip data including, but not be limited to, the Driver, vehicle, date/time of the trip, and origin and destination location with the Electronic Taxi Access System. The data exchanged with the Electronic Taxi Access System shall be transmitted in accordance with data specifications set forth by the Director of Transportation.

(C) The systems used by Dispatch Service Permit Holders shall meet the data standards specified by the Director of Transportation.

API Revision History - October 2023

Version	Date	Editor	Notes
2.0	March 4, 2016	dcorliss	Created
2.1	March 16 th , 2016	dcorliss	Added Driver Status in Telemetry
2.2	April 4, 2016	dcorliss	Added new data elements into trips
2.3	April 14, 2016	dcorliss	Fixed the JSON sample formats for trips and telemetry
2.4	April 18, 2016	dcorliss	Timestamp local made to required field in telemetry API
2.5	December 18, 2018	kkrupp	1) Changed “optional” fields to “required”, going forward 2) Taxi company “name” to be used for normal data transmission, replacement month files, and for sending test data (via normal transmission method)
3.0	March 2020	SFMTA TAMS Data Team, Sean Cunningham, Luke Armbruster	Upgrade to POST AND GET; align (where appropriate) with LA MDS standard; new fields
3.1	July 2021	mwang	Add new required data fields
3.2	June 2022	SFMTA TAMS Data Team, IT Data Services	Incorporate feedback from dispatch companies; field standardization; incorporate transportation code changes; OAuth2 implementation; add device_id
3.2.1	June 21, 2023	mwang, bjessup	Add test vehicle placard number assignment; update OAuth2
3.2.2	October 23, 2023	SFMTA TAMS	Update data transmission standards. Prepare for MDS 2.0 Transition

Authorization

SFMTA Taxi Services shall be secured using OAuth 2.0 Authentication. See the OAuth2 Authentication Instructions section below for details.

Taxi Trips API 3.2.2

Introduction

The Taxi Trip API allows you to transmit one or more Trip records to the SFMTA Electronic Access Taxi System. Trip records must be sent in real-time upon the completion of each trip and include all the required fields as described below.

Request

Method	URL	Environment
POST	https://stageservices.sfmta.com/taxi/api/3/TaxiTrips/	Test
POST	https://services.sfmta.com/taxi/api/3/TaxiTrips/	Production

Request Header

Attribute	Value	Description
Accept	application/json	
Content-Type	application/json	
Authorization	Bearer <token>	See authentication section below for details

New content is labeled as “new field”. Changed fields from the API 2.5 version are noted as “formerly:”.

Payload: { "trips": [] }, an array of objects with the following structure

Field	Description	Required	Type
provider_id (new field)	UUID (“Universal User ID” that is associated with the provider).	Y	String
taxi_company_id (new field)	UUID (“Universal User ID” that is associated with the taxi company).	Y	String
vehicle_id (changed field)	VIN (Vehicle Identification Number) of the taxicab.	Y	String
vehicle_placard_number (new field)	The number painted on the taxicab. AKA: Medallion ID	Y	String A four-digit number such as ‘1824’, ‘0091’, ‘5003’ For testing purposes, each dispatch company should

			use designated test number: Autocab – ‘1999’ CMT – ‘1998’ Flywheel – ‘1997’ Curb – ‘1996’
license_plate (new field)	License plate number of the taxicab.	Y	String
trip_id (formerly: TripNumber)	The id from the taxi company or payment provider to be used for troubleshooting purposes. Could be number that is on transaction receipt or from provider’s data base. Not to be generated by any on-board taxicab equipment or software.	Y	String
device_id (new field)	Identifier indicating the unique device on the vehicle (e.g. hard or soft meter) that transmitted the data.	Y	String
hail_type (formerly: Trip_Type)	How the trip was ordered: street hail, dispatched, names of approved mobile apps.	Y	String Values allowed: (Traditional) - “street”, - “dispatch”, - “curbstand”; (Mobile App) - “yo_taxi_app”, - “flywheel_app”, - “arro_app”, - “curb_app”; (3rd Party Originator) - “flywheel_uber”, - “yo_taxi_uber”, - “arro_uber”, - “curb_uber”
operator_id (formerly: Driver_ID)	The driver's license number. Must be in valid CA DMV format. Out of state driver’s licenses will not be accepted.	Y	String
start_time_milliseconds (formerly: Start_Time_Local)	The date and time when the meter was engaged, in integer milliseconds since Unix epoch.	Y	Number

end_time_milliseconds (formerly: End_Time_Local)	The date and time when the meter was disengaged, in integer milliseconds since Unix epoch.	Y	Number
pickup_location_address	The address of the pickup location, including street address, city, state, zip.	Y	String
pickup_location_latitude	The geo latitude of the pickup location.	Y	Number 5 digits of precision Example: 37.77636
pickup_location_longitude	The geo longitude of the pickup location.	Y	Number 5 digits of precision Example: -123.77636
Drop_off_location_address	The address of the pickup location, including street address, city, state, zip.	Y	String (text)
Drop_off_location_latitude	The geo latitude of the drop-off location.	Y	Number 5 digits of precision Example: 37.77636
Drop_off_location_longitude	The geo longitude of the drop-off location.	Y	Number 5 digits of precision Example: -123.77636
passenger_count	The number of passengers.	Y	Number
is_wheelchair_transported	A flag indicating whether a wheelchair was transported.	Y	String Values allowed: - "T" (True) - "F" (False)
total_fare_amount (new field)	Total fare for the trip, including all tolls, tips, fees, extras, flag drop, and meter amount	Y	Number
meter_fare (new field)	Cost to the customer for the trip, as reported by the meter (excluding tips, fees, tolls, extra amounts). For upfront priced trips, insert what meter rate would be.	Y	Number
upfront_pricing (new field)	Agreed upon rate that should not change based on the meter, (excluding tips, fees, tolls, extra amounts).	N	Number. Leave NULL if not applicable.

promo_rate (new field)	Promotional rate.	N	Number. Ex. Yellow has \$35 SFO flat rate promotion. Leave it NULL if not applicable.
fare_type (new possible values)	Indicator of which rate was charged. Options are Meter Fare (time, distance, flag drop), Upfront Pricing, Promo Rate.	Y	String Values allowed: - meter_fare - upfront_pricing - promo_rate
tolls	Sum of any and all tolls charged for the trip, such as bridge tolls.	N	Number. Leave NULL if not applicable.
rate_code_id (new field)	Indicator of what multiplier was applied to calculate the fare. (1) Meter Fare: based on meter; (2) Flat Rate per person in Shared Rides; (3) Out-of-Town Trips: metered rate *150% for trip that goes 15 miles or more outside of boundaries; (4) Deduction for Time While Disabled: no charge for time while vehicle is disabled; (5) Establishment of Upfront Fare Pilot Program; (6) Promo Rate is any rate less than the other rates.	Y	String Values allowed: - meter_fare - shared - out_of_town - disabled - upfront_pricing - promo_rate
sfo_exit_fee (new field)	Fee paid by customer to exit SFO.	N	Number Leave it NULL if not applicable.
flag_drop_amount (new field)	Amount from the meter that results from “flag drop”. This flag drop amount is included in meter_fare and is not to be added in total fare calculation.	Y	Number
other_fees (formerly: Fees)	Amount of any fees charged to the customer. Includes baggage fees, cleaning fee. Excludes sfo_exit_fee.	N	Number Leave NULL if not applicable.
tip	Amount of tip paid by the customer.	Y	Number Enter 0 if no tip (do not leave NULL)

extra_amount (new field)	Extra amounts charged to the customer. SFMTA does not collect this field at this time. For future use.	N	Number Leave NULL if not applicable.
payment_type (changed field)	How the trip was paid. Credit Card is using the in-taxi payment equipment. Third Party-Originated trips use Mobile.	Y	String Values allowed: - cash - credit_card - mobile - voucher - paratransit - no_payment - test (<i>for test data</i>)
trip_duration_milliseconds (formerly: MeterTripTime)	Trip time as reported in integer milliseconds.	Y	Number
trip_distance_meters (formerly: TripDistance)	The trip distance reported in meters, rounded to the nearest whole meter.	Y	Number
fare_time_milliseconds (new field)	The fare time reported in integer milliseconds. This time accumulates when vehicle travels over 12MPH.	Y	Number
wait_time_milliseconds (new field)	The wait time reported in integer milliseconds. This time accumulates when vehicle travels under 12MPH.	Y	Number
publication_time (new field)	Publication time (in integer milliseconds since Unix epoch) is the time that provider posts to SFMTA	Y	Number

In addition to the required fields above, any valid JSON may be sent, as long as the values are of data type *String* or *Number*.

Request Example

```
{
  "trips": [ {
    "provider_id": " e714f168-ce56-4b41-81b7-0b6a4bd26128",
    "taxi_company_id": " 07a25fe6-d0be-11e8-a8d5-f2801f1b9fd1",
    "vehicle_id": "19XFB4F39EE200589",
    "vehicle_placard_number": "1811",
    "license_plate": "KP20091",
    "trip_id": " c848a5c0-2904-4f96-954a-77cfddcfed4b",
    "device_id": "15505",
    "hail_type": "street",
    "operator_id": "B5471749",
    "start_time_milliseconds": 1622609585000,
    "end_time_milliseconds": 1622609945000,
    "pickup_location_address": "123 Main Street, San Francisco, CA, 94102",
    "pickup_location_latitude": 37.56565,
    "pickup_location_longitude": -123.56565,
    "drop_off_location_address": "1 S Van Ness Ave, San Francisco, CA, 94102",
    "drop_off_location_latitude": 37.56565,
    "drop_off_location_longitude": -123.56565,
    "passenger_count": 2,
    "is_wheelchair_transport": "F",
    "total_fare_amount": 20.00,
    "meter_fare": 15.00,
    "upfront_pricing": ,
    "promo_rate": ,
    "fare_type": "meter_fare",
    "tolls": 2.00,
    "rate_code_id": "meter_fare",
    "sfo_exit_fee": ,
    "flag_drop_amount": 4.15,
    "other_fees": ,
    "tip": 3.00,
    "extra_amount": ,
    "payment_type": "credit_card",
    "trip_duration_milliseconds": 360000
    "trip_distance_meters": 5000,
    "fare_time_milliseconds": 360000
    "wait_time_milliseconds": 0,
    "publication_time": 1622609945000
  } ]
}
```

Response

Request message is either accepted which is indicated by a HTTP Code 202.

If the request is not accepted, a non-200 response code will be returned.

Taxi Telemetry API 3.2.2

Introduction

The Telemetry API allows you to transmit one or more Telemetry records to the SFMTA Electronic Access Taxi System. Telemetry records log the GPS location of taxi vehicles at successive moments in time.

Telemetry records must include all of the required fields described below.

Telemetry records must be transmitted in real-time from every vehicle throughout the entire period the vehicle is in operation and on-duty. The location of each vehicle must be transmitted in latitude and longitude GPS coordinates with a precision of at least 5 decimal places.

Telemetry records must be transmitted from each vehicle with a frequency of 6-8 seconds on average. The transmission frequency must remain steady without fluctuating beyond the range of 3-12 seconds throughout the entire period of transmission.

Telemetry records must be archived by the Dispatch Service Permit Holder for at least one month (30 days) after the date of transmission and be available for retransmission upon request.

Request

Method	URL	Environment
POST	https://stageservices.sfmta.com/taxi/api/3/Telemetries/	Test
POST	https://services.sfmta.com/taxi/api/3/Telemetries/	Production

Request Header

Attribute	Value	Description
Accept	application/json	
Content-Type	application/json	
Authorization	Bearer <token>	<i>See Authentication section below for details.</i>

Payload: { "telemetry": [] }, an array of objects with the following structure

Field	Description	Required	Type
provider_id (new field)	UUID ("Universal User ID" that is associated with the provider).	Y	String
taxi_company_id (new field)	UUID ("Universal User ID" that is associated with the taxi company).	Y	String
vehicle_id (changed field)	VIN (Vehicle Identification Number) of the taxicab.	Y	String
vehicle_placard_number (new field)	The number painted on the taxicab. AKA: Medallion ID	Y	String A four-digit number such as 0001, 1824, 5003.

operator_id (formerly: Driver_License)	The driver's license number. Must be in valid CA DMV format. Out of state driver's licenses will not be accepted.	Y	String
trip_id (new field)	The id from the taxi company or payment provider to be used for troubleshooting purposes. Could be number that is on transaction receipt or from provider's data base. Not to be generated by any on-board taxicab equipment or software. Only for telemetry events where the vehicle status is hired.	N	String Leave NULL if not applicable.
device_id (new field)	Identifier indicating the unique device on the vehicle (hard or soft meter) that transmitted the data.	Y	String
driver_status	Indicates if this taxicab telemetry event represents the start of a driver shift, continuation of current shift, or end of shift.	Y	Number Values allowed: - 1 (Starting Shift) - 2 (On Shift) - 3 (Ending Shift)
latitude	The geographic latitude of the current location of the taxicab trip.	Y	Number 5 digits of precision Example: 37.77636
longitude	The geographic longitude of the current location of the taxicab trip.	Y	Number 5 digits of precision Example: -122.4353
vehicle_status	The taxicab status	Y	Number Values allowed: - 1 (Off Duty) - 2 (Available) - 3 (Hired)
event_time_milliseconds (formerly: Time_Stamp_Local)	Date and time vehicle reported this location, in integer milliseconds since Unix epoch.	Y	Number
publication_time (new field)	Publication time (in integer milliseconds since Unix epoch) is the time that provider posts to SFMTA	Y	Number

In addition to the required fields above, any valid JSON may be sent, as long as the values are of data type *String* or *Number*.

Request Example

```
{
  "telemetry" : [ {
    "provider_id" : " e714f168-ce56-4b41-81b7-0b6a4bd26128",
    "taxi_company_id" : " 07a25fe6-d0be-11e8-a8d5-f2801f1b9fd1",
    "vehicle_id" : "19XFB4F39EE200589",
    "vehicle_placard_number" : "1811",
    "operator_id" : "B5471749",
    "trip_id" : "848a5c0-2904-4f96-954a-77cfddcfed4b",
    "device_id" : "15505",
    "driver_status" : 2,
    "latitude" : 37.5656564,
    "longitude" : -123.5655667,
    "vehicle_status" : 3,
    "event_time_milliseconds" : 1622609945000,
    "publication_time" : 1622609945000
  } ]
}
```

Response

Request message is either accepted which is indicated by a HTTP Code 202.

If the request is not accepted, a non-200 response code will be returned.