



6. Batched submissions will no longer be accepted. All data should be submitted unbatched within ten minutes of occurrence.

7. If requested, operators must be able to provide the SFMTA with at least 30 days of historical telemetry data from the date of the request. Historical telemetry data must contain all of the same fields and data points as real time submissions.

Version: 4.0

Date: April 13, 2018

Version	Date	Editor	Notes
4.0	April 13, 2017	fnapolitan	Updated 2017 requirements for the operator reapplication process in 2018

Stop List API Introduction

The Stop List is a simple repository of the current stops allowed as a part of the Commuter Shuttle Program. These are intended to be a reference for shuttle providers to supply the correct stop ids when posting stop events to the SFMTA. The list is comprehensive and lists all stops used by any providers.

Authorization

The allowed stops API does not require authorization for a GET.

Request

Method	URL	HTTP Version
GET	http(s)://services.sfmta.com/shuttle/api/AllowedStops	1.1

Request Header

Attribute	Value	Description
Accept	application/json	
Content-Type	application/json	

Parameters

Field	Description	Required	Type

Request Example

Response

Fields

Field	Description	Required	Type
-------	-------------	----------	------



StopId	The stop id, normally a Muni stop identifier as used by the SFMTA	Y	Number
StopDescription	The description of the stop	N	String
StopLocationLatitude	The geo latitude of the stop id location in WGS84 latitude	Y	Number (7 digits of precision) Example: 37.776365
StopLocationLongitude	The geo longitude of the stop id location in WGS84 longitude	Y	Number (7 digits of precision) Example: -127.453544

Example

```
{
  "Success" : "True",
  "Stops" : {
    "Stop" : [ {
      "StopId" : 3536,
      "StopDescription" : "30th St&Church St SW corner, flag stop",
      "StopLocationLatitude" : 37.74204,
      "StopLocationLongitude" : -122.42676
    }, {
      "StopId" : 3543,
      "StopDescription" : "30th St&Sanchez St NE corner, near-side stop",
      "StopLocationLatitude" : 37.74202,
      "StopLocationLongitude" : -122.4286
    } ] }
}
```

Telemetry & Stop Events API

Introduction

The purpose of the Telemetry & Stop Events API is to allow shuttle vendors to send the SFMTA a continuous set of GPS location data (latitude and longitude) for the duration of a vehicle's operating time within the boundaries San Francisco. Events should be sent in near real-time, so each payload can only contain one event. The frequency of these GPS points must be every 5 seconds or less. The LocationLatitude and LocationLongitude are required fields for all transactions sent.

Shuttle vendors must provide stop information each time a vehicle stops for the purposes of boarding or alighting. This data should be recorded in the VehicleStatus field. When a vehicle is at a stop actively loading or unloading, VehicleStatus should be 2 (At a Stop). At all other times Vehicle Status should be 1 (On Route).



Listed below are acceptable methodologies for providing stop data:

1. Deployment and release of the parking brake
2. Automated passenger counter
3. Door opening and door closing
4. Rider badge tagging on
5. Geofencing
 - a. Note: if this methodology is selected an explanation of what parameters will be used to avoid false detections and accurately capture all stop events must be provided

Records of stop events made at a zone within the Commuter Shuttle Program’s network should include that zone’s designated stop ID value in the Stop ID field. In cases where the stop is made at a location without a Stop ID (either a temporary stop or a stop not located in the Commuter Shuttle Program network) the Stop ID 9999 should be selected.

In addition to the required fields, any valid JSON may be sent, as long as the values are of data type *String*, *Number*, or *Boolean*. Dates are also supported if formatted correctly and sent as a *String*.

Exceptions

The frequency of GPS points may be every 15 seconds or fewer for the following permittees:

1. Permittees providing only intracity service (service only within San Francisco); or
2. Permittees with 15 or fewer stop events per day.

All other permittees must transmit GPS data every 5 seconds or fewer on all their routes.

Authorization

The API uses Basic Authentication. To use the API, you must have a valid username and password in the HTTP header which the SFMTA will provide to you once you are granted a permit to participate in the program.

Request

Method	URL	HTTP Version
POST	https://services.sfmta.com/shuttle/api/Telemetry/	1.1

Request Header

Attribute	Value	Description
Accept	application/json	
Content-Type	application/json	
Authorization	Basic	

Parameters

Field	Description	Required	Type
TechProviderId	Identification number of technology provider as assigned by SFMTA	Y	Number



ShuttleCompanyId	Identification number of shuttle provider as assigned by SFMTA.	Y	Number
VehiclePlacardNum	The placard number issued to each vehicle. Must be unique for each vehicle.	Y	String (25 characters)
LicensePlateNum	License plate number issued to each vehicle.	Y	String (7 characters)
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	Y (when VehicleStatus is 2)	Number
VehicleStatus	The vehicle status	Y	Number (Options: 1 – On Route 2 – At a Stop)
LocationLatitude	The geo latitude of the vehicle.	Y	Number (7 digits of precision) Example: 37.7763652
LocationLongitude	The geo longitude of the vehicle.	Y	Number (7 digits of precision) Example: 122.435387
TimeStampLocal	Date and time vehicle reported this location	Y	DateTime
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. <i>For example, 00_Mission_AM</i>	N	String (50 characters)
RouteType	Inter or intracity route	N	Number (Options: 1 – Intercity 2 – Intracity)



RunID	Unique number for each run	N	Number
RunOccupany	Current number of passengers onboard the vehicle during the trip.	N	Number

Request Example

```
{
  "TechProviderId" : 130,
  "ShuttleCompanyId" : "99",
  "VehiclePlacardNum" : "99-1234",
  "LicensePlateNum" : "XYZ1234",
  "StopId" : 5001,
  "VehicleStatus" : 1,
  "LocationLatitude" : 37.7739721,
  "LocationLongitude" : -122.4312456,
  "TimeStampLocal" : "2016-01-06T09:53:05.296",
  "RouteIdentifier" : "00_Mission_AM",
  "RouteType": 1,
  "RunID" : 123,
  "RunOccupancy" 1
}
```

Response

Fields

Field	Description	Required	Type
Success	Indicates if the request was process successfully.	Y	String
Message	Description of failure	N	String

Example

Success Response

```
{
  "Success" : "True",
  "Message": "Payload received",
  "PayloadId": 1
}
```

Error Response

```
{
  "RestFaultElement": {
```



```
"code": "invalidVariables",  
"summary": "Unknown Shuttle Company Id"  }  
}
```

Appendix A: Authentication Sample

.Net C# Sample Code

```
WebRequest request = WebRequest.Create(url);  
request.Method = "POST";  
request.ContentType = "application/json";  
String encoded = Convert.ToBase64String(  
    Encoding.GetEncoding("ISO-8859-1").GetBytes("<username>" + ":" + "<password>"));  
request.Headers[HttpRequestHeader.Authorization] = "Basic " + encoded;
```

Or

```
WebRequest request = WebRequest.Create(url);  
request.Method = "POST";  
request.ContentType = "application/json";  
request.Credentials = new NetworkCredential(<username>, <password>);
```