Cellocator Open Protocol Standards

Developer's Guide

Proprietary and Confidential

Version 1.0

Revised and Updated: 5 January 2022

CELLOCATOR - A POWERFLEET® BRAND. 14 HAMELACHA ST. ROSH HA'AYIN 4809133, ISRAEL • TEL: 972-3-5723111 • WWW.CELLOCATOR.COM

Legal Notices

IMPORTANT

1. All legal terms and safety and operating instructions should be read thoroughly before the product accompanying this document is installed and operated.
2. This document should be retained for future reference.
3. Attachments, accessories or peripheral devices not supplied or recommended in writing by PowerFleet Inc. May be hazardous and/or may cause damage to the product and should not, in any circumstances, be used or combined with the product.

General

The product accompanying this document is not designated for and should not be used in life support appliances, devices, machines or other systems of any sort where any malfunction of the product can reasonably be expected to result in injury or death. Customers of PowerFleet Inc. using, integrating, and/or selling the product for use in such applications do so at their own risk and agree to fully indemnify PowerFleet Inc. For any resulting loss or damages.

Warranty Exceptions and Disclaimers

PowerFleet Inc. shall bear no responsibility and shall have no obligation under the foregoing limited warranty for any damages resulting from normal wear and tear, the cost of obtaining substitute products, or any defect that is (i) discovered by purchaser during the warranty period but purchaser does not notify PowerFleet Inc. until after the end of the warranty period, (ii) caused by any accident, force majeure, misuse, abuse, handling or testing, improper installation or unauthorized repair or modification of the product, (iii) caused by use of any software not supplied by PowerFleet Inc., or by use of the product other than in accordance with its documentation, or (iv) the result of electrostatic discharge, electrical surge, fire, flood or similar causes. Unless otherwise provided in a written agreement between the purchaser and PowerFleet Inc., the purchaser shall be solely responsible for the proper configuration, testing and verification of the product prior to deployment in the field.

POWERFLEET INC.'S SOLE RESPONSIBILITY AND PURCHASER'S SOLE REMEDY UNDER THIS LIMITED WARRANTY SHALL BE TO REPAIR OR REPLACE THE PRODUCT HARDWARE, SOFTWARE OR SOFTWARE MEDIA (OR IF REPAIR OR REPLACEMENT IS NOT POSSIBLE, OBTAIN A REFUND OF THE PURCHASE PRICE) AS PROVIDED ABOVE. POWERFLEET INC. EXPRESSLY DISCLAIMS ALL OTHER WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, SATISFACTORY PERFORMANCE AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL POWERFLEET INC. BE LIABLE FOR ANY INDIRECT, SPECIAL, EXEMPLARY, INCIDENTAL OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOSS OR INTERRUPTION OF USE, DATA, REVENUES OR PROFITS) RESULTING FROM A BREACH OF THIS WARRANTY OR BASED ON ANY OTHER LEGAL THEORY, EVEN IF POWERFLEET INC. HAS BEEN ADVISED OF THE POSSIBILITY OR LIKELIHOOD OF SUCH DAMAGES.

Intellectual Property

Table of Contents

List of Tables

List of Figures

1 Introduction

This document is for developers who want to integrate their Cellocator devices into their hosted application using the Cellocator Open Protocol Standards (OPS).

The following diagram depicts the relationship and flow of data within the OPS ecosystem:

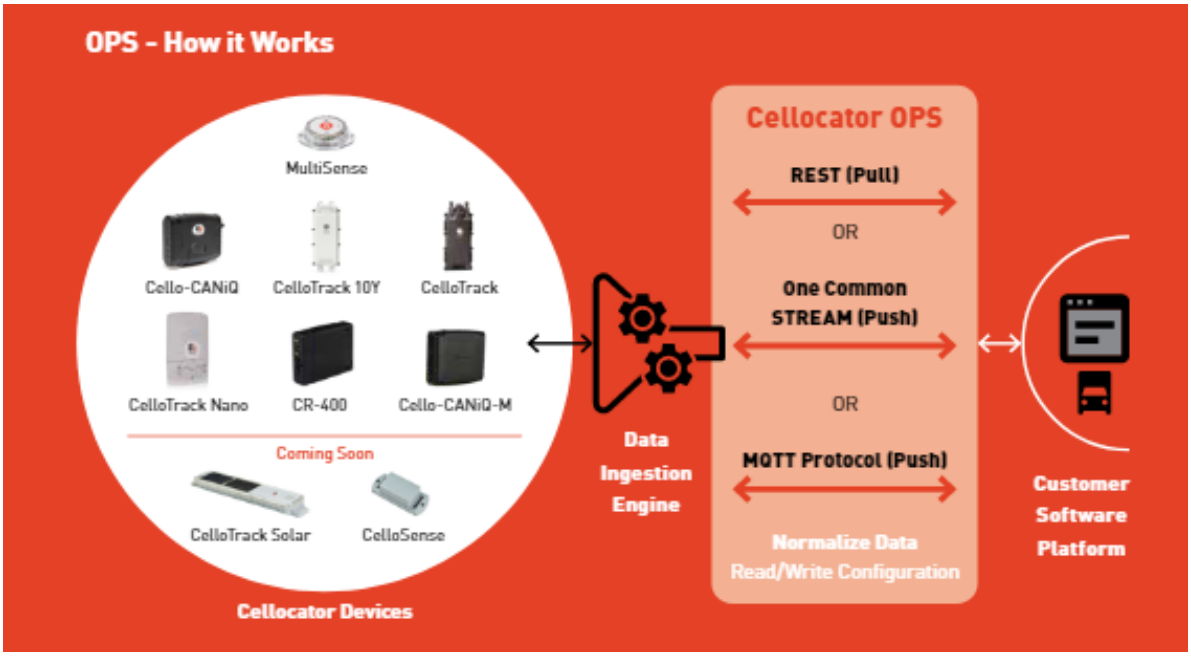

Figure 1 – How OPS Works

Data from the Cellocator devices can be made available in many ways, depending on your requirements. You can choose from the following options:

- **REST API:** Make pull requests for all devices you own, or a single device, for a specified period of time. Note that this option is available by default.
- **MQTT API:** Create your own client connection with our MQTT broker and subscribe to specific topics. Note that this option is available by default.
- **MQTT Broker:** Host your own MQTT broker and stream all data directly from our broker to yours.
- **HTTP POST:** Stream all data to your web server.
- **AWS IoT:** Stream all data to your AWS cloud solution.
- **Azure IoT:** Stream all data to your Azure cloud solution.
- **Google IoT:** Stream all data to your Google cloud solution.

1.1 References and Bibliography

No.	Document Name
1	Cellocator Cello-CANiQ Programming Manual
2	Cellocator Wireless Communication Protocol
3	Cellocator Serial Communication Protocol
4	Cellocator CelloTrack Nano Programmer Manual

Table 1 – References

1.2 List of Changes

Version	Change	Remarks	Date Approved
1.0	First Draft		

Table 2 - List of Changes

2 Getting Started

All account administration is handled by the Cellocator Customer Service team. Device serial numbers must be provided to the Customer Service team to be processed and provisioned to your account. Once your account has been created and devices are provisioned in your account, you should receive a REST/MQTT API token from the Customer Service team that will support the GET methods described in the REST API section, as well as subscribing to the topics outlined in the MQTT section.

If you require a streaming option (not using REST or MQTT API), review the streaming options section below. Each streaming option has its own dedicated section in this document that provides you with the necessary instructions for setting up these integrations.

2.1 Streaming Options

- HTTP Post Streaming – Send HTTP requests to an arbitrary URL containing the JSON payload of device messages.
- MQTT Broker Streaming - Publishes messages directly to your existing MQTT broker.
- Azure IoT – Forward device messages directly to your existing Azure IoT Hub solution.
- AWS IoT – Forward device messages directly to your existing AWS IoT Thing solution.
- Google IoT – Forward device messages directly to your existing Google Cloud IoT Core solution (using Google Cloud Pub/Sub and Google Big Table).

2.2 Definitions

The following fields are used throughout the document and are defined in the table below.

Field	Description
id	The unique primary key for each device. This field can be used to make get requests.
ident	The serial number of the device. This field is not changeable and can be used to make get requests.
name	The serial number of the device. This field can be set to any string value during initial device provisioning. Contact customer service for more information.
timestamp	An epoch formatted timestamp.
dev-selector	A string that defines the query to be used for returning data for a device or list of devices (when using REST API). In most cases, "all" can be used to return data for all devices within the account. Queries can be combined using csv format and will apply a logical AND condition. Examples: /gw/devices/all — returns all devices /gw/devices/1,2,3 — returns devices with id values 1,2,3 /gw/devices/configuration.ident=2*9 — returns devices with ident starting with 2 and ending with 9

Table 3 – Field Definitions

3 REST API

The REST API methods described below represent a collection of APIs that can be used to retrieve data periodically. Requests are limited to once per second and should not be abused or can be temporarily denied service for a period of time.

- **Get Devices**
- **Get Telemetry**
- **Get Device Messages**

3.1 Get Devices

Returns a collection of devices matching filter parameters. Use the special keyword `"all"` to retrieve all devices.

3.1.1 *Execution*

```
curl -X GET  --header 'Authorization: FlespiToken XXXXXXXXX'
'https://flespi.io/gw/devices/{dev-selector}'
```

3.1.2 *Sample Response*

```
{
    "result": [
        {
            "messages_ttl": 1209600,
            "device_type_id": 35,
            "id": 1,
            "name": "vin1234",
            "configuration": {
                "ident": "1234567",
                "settings_polling": "once"
            }
}
```

3.1.3 *Notes*

`{dev-selector}` supports the following options:

- `"all"` – No filter applied, return all devices.
- `"configuration.ident=[serial#]"` – returns all devices containing value of [serial#] in the ident field. Use of wildcard (*) optional at end of string.
- `"1,2,3…"` – choose elements by "id" comma separated list.

3.2 Get Telemetry

Get the latest values of parameters for a specified device.

3.2.1 *Execution*

```
curl -X GET  --header 'Authorization: FlespiToken XXXXXXXXX'
'https://flespi.io/gw/devices/{dev-selector}/telemetry/{telemetry-
selector}'
```

3.2.2 *Sample Response*


```
{
    "result": [
        {
            "id": 2115897,
            "telemetry": {
                "battery.level": {
                    "ts": 1631318271,
                    "value": 71
                },
                "device.id": {
                    "ts": 1631318271,
                    "value": 2115897
         },
                "position": {
                    "ts": 1631318271,
                    "value": {
                        "altitude": 1.8,
                        "direction": 2.692902,
                        "hdop": 0,
                        "latitude": 25.738872,
                        "longitude": -80.261874,
                        "satellites": 6,
                        "speed": 1
                    }
                },
                "position.altitude": {
                    "ts": 1631318271,
                    "value": 1.8
                },
                "position.direction": {
                    "ts": 1631318271,
                    "value": 2.692902
            }
        }
    ]
}
```

3.2.3 *Notes*

`{dev-selector}` supports the following options:

- `"all"` – No filter applied, return all devices.
- `"configuration.ident=[serial#]"` – returns all devices containing a value of [serial#] in the ident field. Use of wildcard (*) optional within string.
- `"1,2,3…"` – choose elements by "id" comma separated list.

`{telemetry-selector}` supports the following options:

- `"all"` – choose all telemetry fields available.
- `"position.latitude,position.longitude,…"` – choose comma separated field list by name.

3.3 Get Device Messages

Get specified device messages.

This request allows for an optional query parameter. When no parameters are provided, this execution will return all device messages. Response is sorted by time; if you request messages from multiple devices simultaneously, they are not sorted by device.

3.3.1 *Execution*

```
curl -X GET  --header 'Authorization: FlespiToken XXXXXXXXX'
'https://flespi.io/gw/devices/{dev-selector}/messages'
```

A Filter parameter can be used to select part of a message. It is represented by the options list. Below is the full data option list schema supported in the body, used to return specified messages:

```
{ // data
count : integer,
// Default: 10000000
/*
limit number of messages to be returned

*/
/*
Restrictions:
Minimum: 1
Maximum: 10000000
*/
fields : string,
/*
CSV-formatted list of parameters to be returned in response message list
in form "timestamp,can,position.*". If message does not contain all
specified fields, it will be skipped.

*/
/*
```

```
Restrictions:
Maximum length: 4096
*/
filter : string,
/*
select only messages that correspond to the supplied filter

*/
/*
Restrictions:
Maximum length: 1024
*/
from : number,
/*
UNIX timestamp of left time border for messages. Use 0 to ignore left bound.

*/
/*
Restrictions: */
generalize : integer,
/*
Generalization interval. Only timestamps that are multiple to specified
generalization interval will be returned with recalculated parameter
values according to specified generalization method. It may be possible
that first returned timestamp will be less than specified 'from'
timestamp if 'from' timestamp is not a multiple of generalization
interval.
All numerical parameter values withing generalization interval will be
calculated into one value according to specified method. For non-
numerical parameters first encountered value will be returned.

*/
/*
Restrictions:
Minimum: 60
Maximum: 1000000
*/
method : string,
reverse : boolean,
/*
time sorting parameter (true/false), use false to list messages from end
to beginning

*/
to : number
/*
UNIX timestamp of right time border for messages. Use 0 to ignore right
bound.

*/
/*
Restrictions: */
}
```

3.3.2 *Sample Execution*

Below is an example of an execution that can be used to pull back location related fields, ident, and time for messages containing location data that was created between 1638487671 and 1638587671 and has gps.mode.1 = 4 and gps.mode.2 = 2.

```
curl --location --request GET
'https://flespi.io/gw/devices/configuration.ident=2447532/messages' \
--header 'Authorization: FlespiToken XXXXXXXXX'  \
--header 'Content-Type: application/json' \
--data-raw '{"from":1638487671, "to":1638587671,
"fields":"ident,timestamp,position.*",
"filter":"gps.mode.1=4,gps.mode.2=2"}'
```

3.3.3 *Sample Response*


```
{
    "result": [
        {
            "ident": "2447532",
            "position.altitude": 26,
            "position.direction": 0,
            "position.latitude": 25.73893,
            "position.longitude": -80.261978,
            "position.satellites": 4,
            "position.speed": 0,
            "timestamp": 1638487671
        }, {
            "ident": "2447532",
            "position.altitude": 6,
            "position.direction": 122.842151,
            "position.hdop": 3,
            "position.latitude": 25.738882,
            "position.longitude": -80.261956,
            "position.satellites": 15,
            "position.speed": 1,
            "timestamp": 1638568625
        }
    ]
}
```


3.3.4 *Notes*

- `{dev-selector}` supports the following options:
  - `"all"` – No filter applied, return all devices.
  - `"configuration.ident=[serial#]"` – returns all devices containing value of [serial#] in the ident field. Use of wildcard (*) optional within string.

- Possible operands:
  - numeric comparison (=, !=, <=, >=, <, >)
  - text mask matching: '=' for mask match, '!=' for mask mismatch, '~' for case insensitive mask match.
- Filtering:
  - param1="name",param2!="",param3>=5* will return list of messages with:
    - param1 string value starts with "name"-symbols.
    - existing parameter named "param2" with any value except empty string with param3 numeric value greater or equal to 5.
- You can enumerate parameters with the ',' and '&&' characters to specify that they are required, or enumerate with the '||' character to make match optional.
  - For example: param1=0||param2=0||param3=0

    will return messages with at least one of param1, param2 or param3 equal to 0
  - param1=0 && param2=0 && param3=0

    will return messages where all param1, param2 and param3 equal to 0.

    This is equivalent to param1=0,param2=0,param3=0

4 MQTT API

The MQTT protocol ensures fast and secure message delivery so you receive an event message as soon as the device uploads it. The protocol is a publish and subscribe methodology where devices will publish data and you must instantiate a client session that will subscribe to the data topic to receive those messages.

4.1 Configuration

Host	mqtt.flespi.io
Port	MQTT over TCP: 8883 (SSL) or 1883 (non-SSL) MQTT over WebSockets: 443 (SSL) or 80 (non-SSL)
Versions Supported	3.1, 3.1.1, 5.0
QoS Supported	QoS 0, QoS 1, QoS 2
Authorization	Username – FlespiToken XXXXXXXXXXXXXXXXXX (your token) Password – not used Client ID - any

4.2 Topics

Topic	Description
flespi/message/gw/devices/{device_id}	New messages received by device. {device_id}: Use "+" to subscribe to all devices. Otherwise specify "id" of device or devices (supports comma separated list).
flespi/state/gw/devices/{id}/telemetry/{parameter}	Latest value and modification timestamp for specified parameter(s) and specified device(s). All new received messages by device also generate dedicated publication per each {parameter} into specified topic. {id}: Use "+" to subscribe to all devices. Otherwise specify "id" of device or devices (supports comma separated list). {parameter}: Use "+" to subscribe to all field updates. Otherwise specify parameter names (supports comma separated list).

4.2.1 *Examples*

- flespi/message/gw/devices/1234567 – All messages for device id 1234567


```
{"channel.id":76381,"device.firmware.version":"38w","device.id":1234567,"device.name":"2000000","device.type.id":35,"gps.mode.1":4,"gps.mode.2":2,"hardware.version.enum":10,"ident":"2000000","message.buffered.status":true,"message.type":"11","module.id.2":true,"module.id.25":true,"operator.id":0,"operator.type":30,"peer":"162.244.250.151:33591","pl.signature":545521447,"position.altitude":187,"position.direction":203.170834,"position.hdop":3,"position.latitude":31.227449,"position.longitude":-
```

```
96.761572,"position.satellites":17,"position.speed":43,"private.status":false,
"protocol.id":20,"server.timestamp":1640212511.906215,"time.valid.status":true
,"timestamp":1640212511,"trigger.cause.enum":1,"variable.data.hex.1":"32088390
","variable.data.hex.10":"4A020301","variable.data.hex.11":"141CD401","variabl
e.data.hex.12":"64647D05","variable.data.hex.13":"DF016404","variable.data.hex
.14":"0020FFF2","variable.data.hex.15":"5E022013","variable.data.hex.16":"A10E
C001","variable.data.hex.17":"00004007","variable.data.hex.18":"62145804","var
iable.data.hex.19":"43030D8E","variable.data.hex.2":"0C8EA12D","variable.data.
hex.20":"9A0E6F0E","variable.data.hex.21":"5B63586D","variable.data.hex.22":"C
E072400","variable.data.hex.23":"5017A802","variable.data.hex.24":"220E01B5","
variable.data.hex.25":"02010040","variable.data.hex.3":"1410013E","variable.da
ta.hex.4":"79011300","variable.data.hex.5":"FFFF0000","variable.data.hex.6":"0
E98701F","variable.data.hex.7":"79020600","variable.data.hex.8":"0000E183","va
riable.data.hex.9":"FF85177D","variable.id.1":16520,"variable.id.10":16515,"va
riable.id.11":16516,"variable.id.12":16517,"variable.id.13":16518,"variable.id
.14":16519,"variable.id.15":16524,"variable.id.16":16527,"variable.id.17":1653
0,"variable.id.18":16531,"variable.id.19":16532,"variable.id.2":16521,"variabl
e.id.20":16533,"variable.id.21":16534,"variable.id.22":16535,"variable.id.23":
16536,"variable.id.24":16528,"variable.id.25":16529,"variable.id.3":16522,"var
iable.id.4":16523,"variable.id.5":16525,"variable.id.6":16526,"variable.id.7":
16512,"variable.id.8":16513,"variable.id.9":16514}
```


- flespi/state/gw/devices/+/telemetry/position,vehicle.odometer – All devices, any update to position fields or odometer.


```
{"altitude":180,"direction":212.911117,"latitude":30.237264,"longitude":-
97.753515,"satellites":16,"speed":12.5}
```


4.2.2 *Notes*

- Quality of Service: Recommended QoS 1 (at least once). This is often a good balance between reliability and speed. See link for more information on this subject.
- Sessions: MQTT connections can be opened with a clean-session flag. In a clean-session mode, subscriptions and undelivered messages are removed as soon as the connection to the broker closes. Non-clean session is persistent — it keeps undelivered messages and subscriptions until the new connection to the broker is established (up to 10 days or 1GB). We strongly recommend the use of non-clean sessions.
- Monitoring: Different options are available for monitoring your sessions. Please see the section on *Monitoring* for more information.

4.3 Monitoring

When using MQTT API, it may become necessary to ensure your session is working as desired and there is no backlog of data for your client waiting to receive. Monitoring the session can assist with this activity and is recommended.

4.3.1 *Session Summary*

This section describes the use of monitoring an MQTT session through an API call. This can be used to help you implement an alert or workflow that will help to reestablish a severed connection.

4.3.1.1 Example

```
curl --location --request GET 'https://flespi.io/mqtt/sessions/+' \
--header 'Authorization: FlespiToken XXXXXXXX'
```

The response will look like this with the "connected" field indicating if the client is currently connected. Note this is in an array format as multiple client connections may exist:

```
{
    "result": [
        {
            "size": 0,
            "ip": "11.2.333.444",
            "id": 4294023833,
            "connected": true,
            "cid": 123456,
            "clean": true,
            "client_id": "tsp-client-connection-name",
            "expires": 0
        }
    ]
}
```

If the response comes back empty, no connections exist. It is recommended that this call be used from a monitoring agent that can alert when the connection is severed (no response or connected=false).

4.3.2 *MQTT State API*

The summary detail works well in most cases, providing insight into connection status, and internal memory allocation, but does not show the queue of unacknowledged messages from the client which often can be helpful in debugging.

The platform does support this information but it must be queried inline within the URL itself and therefore it is recommended to first build the query in plaintext before decoding back to a url encoded string.

4.3.2.1 Available Fields

- puback_queue – delivered messages not yet acknowledged by client
- bytes_out_queue – bytes in memory scheduled to be delivered
- bytes_out – total number of bytes delivered since last session connection
- bytes_in – total number of bytes received since last session connection

4.3.2.2 Format

`https://flespi.io/mqtt/messages/{query}`

The format of the query must be URL encoded text and should follow this format (text shown is not encoded):

`flespi/state/mqtt/sessions/{session-selector}/{field-selector}`

4.3.2.3 Sample Execution: Retrieve the current puback_queue size

```
curl --location --request GET
'https://flespi.io/mqtt/messages/flespi%2Fstate%2Fmqtt%2Fsessions%2F%2B%2F
puback_queue' \
--header 'Authorization: FlespiToken XXXXXX'
```

4.3.2.4 Sample Response

```
{
    "result": [
        {
            "payload": "0",
            "timestamp": 1640288088.537856,
            "type": 2,
            "topic": "flespi/state/mqtt/sessions/2934020141/puback_queue",
            "token_id": 0
        }
    ]
}
```

4.3.2.5 Notes

- `{session-selector}` is the id of the session, which can be obtained by calling the REST session summary API described above. This can also simply be "+" indicating all sessions.
- `{field-selector}` is where the response fields must be specified. It is recommended that only fields mentioned above in the available fields are specified (many others are available but are redundant or not relevant).
- Multiple fields can be specified, comma delimited.
- Payload will contain the value of the field. All other fields can be ignored.

5 HTTP Post Streaming

This section provides details on how to get your account setup for using HTTP Post streaming.

Please review the items below and provide the relevant answers in the form of an email to the Cellocator customer service team with the subject "OPS Account Setup". Please be sure to include the streaming option in the email.

5.1 Setup Questions

- Queue TTL (maximum 7 days): The maximum number of days your messages will be kept in queue for future delivery in the event your service is inaccessible/down.
- Your Web Server URI (please specify http or https).
- Message count limit (default 100): Soft-limit for number of messages to batch in a single request.

5.2 Notes

- As soon as the server replies with an OK (200) the messages are removed from the stream buffer.
- Data will remain in the queue for Queue TTL.
- Message count limit is not guaranteed, but applied whenever possible.

6 MQTT Broker Streaming

This section provides details on how to get your account setup for using MQTT Broker streaming.

Please review the items below and provide the relevant answers in the form of an email to Cellocator customer service team with the subject "OPS Account Setup". Please be sure to include the streaming option in the email.

6.1 Setup Questions

- Queue TTL (maximum 7 days): The maximum number of days your messages will be kept in queue for future delivery in the event your service is inaccessible/down.
- Your Broker URI (please specify http or https).
- SSL (yes or no).
- Client ID.
- Username.
- Password.
- Topics.
- CA (if applicable).
- Certificate (If applicable, crt file content).
- Private key (If applicable, certificate pem key file content).

6.2 Notes

- The stream uses QoS=1 to ensure that the message is removed from the stream buffer as soon as ACK from the broker is received.

7 Azure IoT

This section provides details on how to get your account setup for using Azure IoT streaming.

Please review the items below and provide the relevant answers in the form of an email to the Cellocator customer service team with the subject "OPS Account Setup". Please be sure to include the streaming option in the email.

7.1 Setup Questions

- Queue TTL (maximum 7 days): The maximum number of days your messages will be kept in queue for future delivery in the event your service is inaccessible/down.
- Azure IoT Hub Hostname.
- Azure Device ID.
- Azure Device Policy Key.

8 AWS IoT

This section provides details on how to get your account setup for using AWS IoT streaming.

Please review the items below and provide the relevant answers in the form of an email to the Cellocator customer service team with the subject "OPS Account Setup". Please be sure to include the streaming option in the email.

8.1 Setup Questions

- Queue TTL (maximum 7 days): The maximum number of days your messages will be kept in queue for future delivery in the event your service is inaccessible/down.
- AWS IoT Endpoint.
- Certificate (contents of certificate.pem.crt file).
- Private key (contents of private.pem.key file).

9 Google IoT

This section provides details on how to get your account setup for using Google IoT streaming.

Please review the items below and provide the relevant answers in the form of an email to the Cellocator customer service team with the subject "OPS Account Setup". Please be sure to include the streaming option in the email.

9.1 Setup Questions

- Queue TTL (maximum 7 days): The maximum number of days your messages will be kept in queue for future delivery in the event your service is inaccessible/down.
- Google Cloud Platform Project ID.
- Google Cloud Data Center Location (see https://cloud.google.com/about/locations).
- Google Cloud IoT Registry ID.
- Google Cloud IoT Core Device ID.
- Private Key (contents of rsa_private.pem file).
- Token TTL (maximum of 24hrs for generated JWT).

10 Fields

This section includes a comprehensive list of all Cellocator fields with definitions. Each device will send a subset of these fields. See the **Devices** section for more details on device-specific fields.

Name	Type	Unit	Description
maneuver.abs.state	number		ABS sensor reading for braking event only
sensor.x.acceleration	number	g	Acceleration sensor X-axis value
device.x.acceleration	number	g	Acceleration sensor X-axis value detected by device
sensor.y.acceleration	number	g	Acceleration sensor Y-axis value
device.y.acceleration	number	g	Acceleration sensor Y-axis value detected by device
sensor.z.acceleration	number	g	Acceleration sensor Z-axis value
device.z.acceleration	number	g	Acceleration sensor Z-axis value detected by device
maneuver.x.acceleration.average	number	g	Accelerometer X axes average
maneuver.y.acceleration.average	number	g	Accelerometer Y axes average
report.reason.data	number		Additional transmission data
device.pressure.altitude	number	m	Altitude above sea-level determined by air pressure detected by device
position.altitude	number	meters	Altitude value for position
device.illuminance	number	lux	Ambient light illuminance detected by device
trip.eco.score	number	percentage	A value representing the weighted Eco score of the trip
trip.safety.score	number	percentage	A value representing the weighted safety score of the trip
maneuver.engine.rpm.average	number	rpm	Average engine RPM in maneuver
maneuver.average.speed.green	number	km/h	Average speed in zone Green
maneuver.average.speed.red	number	km/h	Average speed in zone Red
maneuver.average.speed.yellow	number	km/h	Average speed in zone Yellow
trip.average.speed	number	km/h	Average speed of the trip
gsm.bsic	number		Base station identity code
battery.health	number	percentage	Battery health level (SoH)
battery.state.bitmask	number		Battery state bitmask, 8 bits, indexed parameter
battery.temperature	number	celsius	Battery temperature
message.buffered.status	boolean		Black box message
calibrating.status	boolean		Calibrating status
calibration.matrix.index	number		Calibration matrix index, indexed parameter
calibration.phase	number		Calibration phase, 1 or 2
calibration.state.enum	number		Calibration state, 0 - Started, 1 - OK, 2 - Bad Installation, 3 - In progress, 4 - Error
calibration.status	boolean		Calibration status, true - calibrating, false - calibration finished
calibration.step	number		Calibration step, 0 - 10
can.dtc.state.bitmask	number		CAN DTC transmission reason bitmask
crash.direction.enum	number		Crash direction - front, rear, left, right etc.
crash.duration	number	seconds	Crash duration
crash.event	boolean		Crash event detected
crash.id	number		Crash ID
crash.orientation	number		Crash orientation
crash.type	number		Crash type
device.battery.level	number	percentage	Device battery level

Name	Type	Unit	Description
device.io.bitmask	number		Device inputs and outputs status bitmask
private.status	boolean		Device is in private mode
ident	string		Device unique identifier like serial number, IMEI, etc
can.dtc	string		Diagnostic trouble code read from CAN
altitude.difference	number	meters	Difference between the last stable altitude and current filtered one
trip.mileage	number	km	Distance driven since engine start
sensor.door.status	boolean		Door opening status detected by sensor: true - opened, false - closed
driver.id	string		Driver identification card id
driving.status	boolean		Driving status
engine.ignition.status	boolean		Engine ignition status
engine.status	boolean		Engine status
event.category	number		Event category
trigger.cause.enum	number		Event trigger cause: 1 - from low to high, 2 - gtom high to low, 3 - both from low to high and gtom high to low, 4 - equal to threshold
onewire.sensor.serial	string		External 1-Wire sensor serial number in HEX representation
onewire.sensor.temperature	number	celsius	External 1-Wire temperature sensor value
external.alarm.status	boolean		External alarm status
external.memory.type.enum	number		External non-volatile memory type
sensor.battery.level	number	percentage	External sensor battery level
sensor.mac.address	string		External sensor MAC address
device.firmware.version	string		Firmware revision
free.fall.event	boolean		Free fall evemt detected
trip.fuel.consumed	number	liters	Fuel consumed in the last trip
fuel.level	number	percentage	Fuel level percentage
trip.stop.fuel.level	number	percentage	Fuel level upon trip end
trip.start.fuel.level	number	percentage	Fuel level upon trip start
fuel.volume	number	liters	Fuel volume
gnss.status	boolean		GNSS receiver on/off status
maneuver.start.position.valid	boolean		GPS is fixed at the maneuver start moment
maneuver.stop.position.valid	boolean		GPS is fixed at the maneuver stop moment
maneuver.start.latitude	number	degrees	GPS location latitude of the maneuver start moment
maneuver.stop.latitude	number	degrees	GPS location latitude of the maneuver stop moment
maneuver.start.longitude	number	degrees	GPS location longitude of the maneuver start moment
maneuver.stop.longitude	number	degrees	GPS location longitude of the maneuver stop moment
gps.mode	number		GPS mode
gsm.cellid	number		GSM base station ID
gsm.lac	number		GSM location area code
gsm.mcc	string		GSM mobile country code
gsm.mnc	string		GSM mobile network code
gsm.network.roaming.status	boolean		GSM network in roaming mode
hardware.interface.enum	number		Hardware interface and peripherals identifier
hardware.type	string		Hardware type
hardware.version.enum	number		Hardware version enum
harsh.acceleration.event	boolean		Harsh acceleration detected

Name	Type	Unit	Description
harsh.braking.event	boolean		Harsh braking detected
harsh.turn.left.event	boolean		Harsh left turn detected
harsh.turn.right.event	boolean		Harsh right turn detected,
harsh.turn.event	boolean		Harsh turn event detected
position.direction	number	degrees	Heading angle at position detection moment
ibutton.code	string		Hexadecimal code of connected iButton
payload.hex	string		HEX Payload received from device
trip.highway.driving.score	number		Highway driving score of the trip
trip.highway.score.time	number	seconds	Highway driving score time of the trip
position.hdop	number		Horizontal dilution of precision
sensor.humidity	number	percentage	Humidity sensor value, i.e. relative air humidity level
trip.idle.score	number		Idle score of the trip
trip.idle.score.time	number	seconds	Idle score time of the trip
channel.id	number		ID of channel that received a message
device.id	number		ID of device that received a message
device.type.id	number		ID of device type of device that received a message
operator.id	number		ID of operator that triggered the event, refer to Cello-CANiQ Integration Manual
protocol.id	number		ID of protocol
sensor.illuminance	number	lux	Illuminance measured by light sensor
impact.event	boolean		Impact event
acceleration.rms	number		Impact or free-fall acceleration RMS value where each of the axis is in 250µg units
module.id	boolean		Indexed parameter, that shows which modules are present in the message of type 11
position.speed	number	km/h	Instant speed at position detection moment
maneuver.total.fuel.flow	number		Integral of Fuel flow
gsm.sim.iccid	string		Integrated Circuit Card Id of SIM card
peer	string		IP:port from which device connecting to the channel
sensor.package.status.additional	boolean		Last detected package state by the specific MultiSense, received in additional information module: false - close, true - open
sensor.humidity.additional	number	percentage	Last measured humidity by the specific MultiSense, received in additional information module
sensor.illuminance.additional	number	lux	Last measured illuminance (light level) by the specific MultiSense, received in additional information module
sensor.magnet.status.additional	boolean		Last measured magnetic state by the specific MultiSense, received in additional information module: false - magnet not present, true - magten present
sensor.temperature.additional	number	celsius	Last measured temperature by the specific MultiSense, received in additional information module
sensor.x.acceleration.additional	number	g	Last measured X acceleration by the specific MultiSense, received in additional information module
sensor.y.acceleration.additional	number	g	Last measured Y acceleration by the specific MultiSense, received in additional information module

Name	Type	Unit	Description
sensor.z.acceleration.additional	number	g	Last measured Z acceleration by the specific MultiSense, received in additional information module
position.latitude	number	degrees	Latitude coordinate value
position.longitude	number	degrees	Longitude coordinate value
network.signal.rsrp	number	dbm	LTE reference signal received power (RSRP)
sensor.magnet.status	boolean		Magnetic field detected by sensor
mandown.event	boolean		Mandown event detected
maneuver.average.speed	number	km/h	Maneuver average speed
maneuver.duration	number	seconds	Maneuver duration
maneuver.harsh.acceleration.event	boolean		Maneuver extended information: harsh acceleration event
maneuver.harsh.braking.event	boolean		Maneuver extended information: harsh braking event
maneuver.harsh.turn.left.event	boolean		Maneuver extended information: harsh turn left event
maneuver.harsh.turn.right.event	boolean		Maneuver extended information: harsh turn right event
maneuver.max.x.acceleration	number	g	Maneuver extended information: maximum acceleration on the X azis
maneuver.max.y.acceleration	number	g	Maneuver extended information: maximum acceleration on the Y azis
maneuver.overspeeding.event	boolean		Maneuver extended information: overspeeding event
maneuver.harsh.turn.angle	number	degrees	Maneuver extended information: the angle of harsh turn event
maneuver.id	number		Maneuver id
maneuver.max.speed	number	km/h	Maneuver max speed
maneuver.risk.score	number		Maneuver risk value 0-100
maneuver.delta.speed	number	km/h	Maneuver speed delta
maneuver.timestamp	number	seconds	Maneuver start time timestamp
maneuver.type	number		Maneuver type
maneuver.type.enum	string		Maneuver type
maneuver.x.acceleration.max	number	g	Max accelerometer value X in maneuver
maneuver.y.acceleration.max	number	g	Max accelerometer value Y in maneuver
maneuver.z.acceleration.max	number	g	Max accelerometer value Z in maneuver
maneuver.engine.rpm.max	number	rpm	Max engine RPM in maneuver
crash.max.acceleration	number		Maximum acceleration (x or y) during crash
trip.x.acceleration.max	number	g	Maximum accel input on X axis of the trip
trip.y.acceleration.max	number	g	Maximum accel input on Y axis of the trip
trip.z.acceleration.max	number	g	Maximum accel input on Z axis of the trip
maneuver.max.fuel.flow	number		Maximum fuel flow during acceleration events
trip.max.speed	number	km/h	Maximum speed of the trip
timestamp	number	seconds	Message timestamp
message.type	string		Message type
sensor.readings	array		MiltiSense measurement readings: array of JSON objects with "value" and "timestamp" files
sensor.readings.type	number		MiltiSense measurement readings type: 1 - temperature readings, 2 -humidity readings
modem.type.enum	number		Modem type
crash.movement.status	boolean		Movement status associated with crash event: true - crash when driving, false - crash during parking

Name	Type	Unit	Description
sensor.battery.level.additional	number	percentage	MultiSense battery level, received in additional information module
sensor.rssi.additional	number	dBm	MultiSense BLE signal RSSI, received in additional information module
sensor.mac.address.additional	string		MultiSense MAC address, received in additional information module
device.name	string		Name of device that received a message
samples.before.crash	number		Number of acceleration aamples between last GPS before crash and crash
samples.after.crash	number		Number of acceleration samples between crash and next GPS
can.dtc.source	number		Number of bus ECU that reported the DTC
time.after.crash	number	seconds	Number of complete GPS seconds after crash
time.before.crash	number	seconds	Number of complete GPS seconds before crash
can.dtc.number	number		Number of diagnostic trouble codes
engine.on.idling.duration	number	seconds	Number of seconds Engine is on during idling session (engine running)
accelerometer.idling.duration	number	seconds	Number of seconds Idling detection is based on Accelerometer
gnss.idle.duration	number	seconds	Number of seconds Idling detection is based on Valid GNSS position
ignition.on.idling.duration	number	seconds	Number of seconds Ignition is on during idling session (ignition switch is on, Motor is off)
overspeeding.event	boolean		Overspeeding detected
sensor.package.status	boolean		Package status read from external sensor: true - open, false - closed. Indexed parameter, measurement source is used as a parameter index
can.pending.dtc	string		Pending diagnostic trouble code read from CAN
gsm.pcid	number		Physical layer Cell ID in LTE and NR mobile networks
pl.signature	number		PL signature - a unique 32-bit code identifying the PL, refer to Cello-CANiQ Integration Manual
processor.enum	number		Processor family identifier
event.enum	number		Protocol-specific event code
protocol.version	string		Protocol version string
position.satellites	number		Quantity of satellites used to calculate coordinates for given position information
record.seqnum	number		Registered record sequence number
rollover.alarm.status	boolean		Roll over alarm state
rollover.event	boolean		Roll over event detected
row.logging.status	boolean		Row logging status
sensor.battery.dead.status	boolean		Sensor battery dead status, true - battery is dead, false - battely is recovered
sensor.battery.low.status	boolean		Sensor battery status, true - battery low, false - battery ok
sensor.battery.voltage	number	volts	Sensor battery voltage
sensor.connected.status	boolean		Sensor connected status, true - connected, false - not connected
sensor.group.id	number		Sensor group ID
sensor.humidity.alert	boolean		Sensor humidity alert
sensor.type.bitmask	number		Sensor HW type bitmask
sensor.paired.status	boolean		Sensor paired status: true - paired, false - unpaired

Name	Type	Unit	Description
sensor.error.code	number		Sensor problem code
sensor.rssi	number	dBm	Sensor RSSI
sensor.temperature.alert	boolean		Sensor temperature alert
sensor.report.reason	number		Sensor transmission reason code
sensor.zone	number		Sensor zone number
short.idling.duration	number	seconds	Short Idling duration from "Short idling TO" to end of Idling session or until "Long Idling TO"
external.memory	number		Size of external non-volatile memory
internal.memory	number		Size of internal non-volatile memory
program.memory	number		Size of program memory
volatile.memory	number		Size of volatile memory
software.version.enum	number		Software version enum
software.version	string		Software version string
sensor.power.on.status	boolean		Sonser power on status, true - sensor power-up, false - sensor power-down by button
measurement.source	number		Source of measurement
speed.source	string		Source of speed: gps or pulse frequency input
secondary.engine.status	boolean		Standby engine status
gsm.signal.dbm	number	dbm	Strength of GSM signal
device.temperature	number	celsius	Temperature of device
device.temperature.valid	boolean		Temperature readings validity
sensor.temperature	number	celsius	Temperature sensor value
gsm.network.type	string		The type of the mobile network the device is currently registered to
sensor.reading.age	number	seconds	Time elapsed from last sensor value reception
maneuver.time.green	number	seconds	Time in maneuver zone Green
maneuver.time.red	number	seconds	Time in maneuver zone Red
maneuver.time.yellow	number	seconds	Time in maneuver zone Yellow
time.valid.status	boolean		Time is accurate
gsm.timestamp	number	seconds	Timestamp for GSM data
sensor.last.timestamp.additional	number	seconds	Timestamp of last communication from the specific MultiSense, received in additional information module
sensor.last.timestamp	number	seconds	Timestamp of the last communication with the sensor
maneuver.start.timestamp	number	seconds	Timestamp of the maneuver start moment
maneuver.stop.timestamp	number	seconds	Timestamp of the maneuver stop moment
trip.start.timestamp	number	seconds	Timestamp value upon trip start
position.timestamp	number	seconds	Timestamp when coordinates where calculated
server.timestamp	number	seconds	Timestamp when server received a message
trip.movement.time	number	seconds	Time the vehicle has moved from Ignition on to Ignition off event or until driver was replaced
trip.idle.time	number	seconds	Time the vehicle was idle based on GPS and accelerometer information, according to corresponding threshold
vehicle.mileage	number	km	Total calculated mileage
maneuver.total.events.green	number		Total events in maneuver zone Green
maneuver.total.events.red	number		Total events in maneuver zone Red
maneuver.total.events.yellow	number		Total events in maneuver zone Yellow
maneuver.total.severity.enum	number		Total Maneuver Severity: 1 – green, 2-yellow, 3-red
trailer.id	string		Trailer ID
report.reason.enum	string		Transmission reason

Name	Type	Unit	Description
report.reason	number		Transmission reason code
trip.duration	number	seconds	Trip duration
trip.id	number		Trip ID increasing every engine-ON event
operator.type	number		Type of operator that triggered the event, refer to Cello-CANiQ Integration Manual
operation.mode.enum	string		Units operation mode
trip.urban.driving.score	number		Urban driving score of the trip
trip.urban.score.time	number	seconds	Urban driving score time of the trip
baud.rate.enum	number		Used baud-rate enum value
variable.data.hex	string		Variable data in hex format, refer to Cello-CANiQ Integration Manual
variable.id	number		Variable ID, refer to Cello-CANiQ Integration Manual
vehicle.odometer	number		Vehicle mileage in base units configured in device (km, miles, 100m etc)
vehicle.type	number		Vehicle type
ain	number	volts	Voltage on the analog input
work.id	number		Work ID counter

Table 4 – Cellocator Fields

11 Devices

This section includes details on device-specific fields for the following devices:

- **CANiQ**
- **CelloTrack**
- **CelloTrack 10Y**
- **Nano**
- **MultiSense**

11.1 CANiQ

11.1.1 *Message Type 11: Location Data for Type 11*

```
{
channel.id : 1234
device.firmware.version : "38w"
hardware.version.enum : 10
ident : "1234567"
message.buffered.status : true
message.type : "11"
peer : "162.244.250.151:55681"
position.altitude : 274
position.direction : 4.068
position.hdop : 5
position.latitude : 40.456147
position.longitude : -79.931608
position.satellites : 11
position.speed : 1
private.status : false
protocol.id : 20
server.timestamp : 1633114177.452708
time.valid.status : true
timestamp : 1633114176
}
```

11.1.2 *Module 2 (Sub Message of Message Type 11): CAN Bus Data*

```
{
channel.id : 1234
ident : "1234567"
module.id : 2
peer : "162.244.250.151:41762"
pl.signature : 545521447
protocol.id : 20
server.timestamp : 1633113726.908333
timestamp : 1633113724
```

```
variable.data.hex.1 : "CF088C90"
variable.data.hex.10 : "21FD0001"
variable.data.hex.11 : "1400CF01"
variable.data.hex.12 : "69648801"
variable.data.hex.13 : "00FE0DFB"
variable.data.hex.14 : "00C0FF2A"
variable.data.hex.15 : "D2006F0F"
variable.data.hex.16 : "6F0BB000"
variable.data.hex.17 : "00000007"
variable.data.hex.18 : "CC159DFB"
variable.data.hex.19 : "D600019D"
variable.data.hex.2 : "5E2FA12D"
variable.data.hex.20 : "960F8C0F"
variable.data.hex.21 : "585C556A"
variable.data.hex.22 : "860D0C00"
variable.data.hex.23 : "F60D0402"
variable.data.hex.24 : "00000000"
variable.data.hex.25 : "01000000"
variable.data.hex.3 : "17131140"
variable.data.hex.4 : "83000700"
variable.data.hex.5 : "CE0D0000"
variable.data.hex.6 : "75734017"
variable.data.hex.7 : "79013500"
variable.data.hex.8 : "0000E083"
variable.data.hex.9 : "FF970000"
variable.id.1 : 16520
variable.id.10 : 16515
variable.id.11 : 16516
variable.id.12 : 16517
variable.id.13 : 16518
variable.id.14 : 16519
variable.id.15 : 16524
variable.id.16 : 16527
variable.id.17 : 16530
variable.id.18 : 16531
variable.id.19 : 16532
variable.id.2 : 16521
variable.id.20 : 16533
variable.id.21 : 16534
variable.id.22 : 16535
variable.id.23 : 16536
variable.id.24 : 16528
variable.id.25 : 16529
variable.id.3 : 16522
variable.id.4 : 16523
variable.id.5 : 16525
variable.id.6 : 16526
```

```
variable.id.7 : 16512
variable.id.8 : 16513
variable.id.9 : 16514
}
```

11.1.3 *Module ID 25 (Sub Message of Message Type 11): Triggers (Legacy)*

```
{
channel.id : 1234
ident : "1234567"
module.id : 25
operator.id : 30
operator.type : 1
peer : "162.244.250.151:50689"
protocol.id : 20
server.timestamp : 1633113395.159928
timestamp : 1633113394
}
```

11.1.4 *Message Type 0: Generic Events (i.e. Engine On/Engine Off Events/Timer Events)*

```
{
ain.1 : 120
ain.2 : 246
ain.3 : 146
ain.4 : 182
channel.id : 1234
device.io.bitmask.1 : 99
device.io.bitmask.2 : 128
device.io.bitmask.3 : 0
device.io.bitmask.4 : 0
gnss.status : true
gps.mode.1 : 4
gps.mode.2 : 2
gsm.mcc : "310"
gsm.mnc : "410"
gsm.network.roaming.status : false
hardware.type : "0A"
hardware.version.enum : 42
ibutton.code : "000000000000"
ident : "1234567"
message.buffered.status : true
message.type : "0"
modem.type.enum : 1
operation.mode.enum : "standby engine on"
```

```
peer : "162.244.250.151:42242"
position.altitude : 218
position.direction : 71.27595
position.latitude : 40.482956
position.longitude : -79.951095
position.satellites : 11
position.speed : 0.1
private.status : false
protocol.id : 20
protocol.version : "100"
report.reason : 44
report.reason.data : 0
report.reason.enum : "timed event"
server.timestamp : 1633113394.169239
software.version : "38w"
speed.source : "gps"
time.valid.status : true
timestamp : 1633113393
vehicle.odometer : 6
}
```

11.2 CelloTrack

11.2.1 *Message Type 11 Module 41: Tracking Data*

```
{
            "ain.1": 221,
            "ain.2": 0,
            "ain.3": 180,
            "ain.4": 146,
            "channel.id": 71085,
            "device.id": 2260949,
            "device.io.bitmask.1": 12,
            "device.io.bitmask.2": 0,
            "device.io.bitmask.3": 16,
            "device.io.bitmask.4": 0,
            "device.name": "2460648",
            "device.type.id": 37,
            "gnss.status": true,
            "gsm.mcc": "310",
            "gsm.mnc": "410",
            "gsm.network.roaming.status": false,
            "ident": "2460648",
            "module.id": 41,
            "operation.mode.enum": "standby engine off",
            "peer": "162.244.250.197:48582",
            "protocol.id": 20,
```

```
            "protocol.version": "4",
            "report.reason": 70,
            "report.reason.data": 0,
            "server.timestamp": 1634659262.951237,
            "speed.source": "gps",
            "time.valid.status": true,
            "timestamp": 1634659139,
            "vehicle.odometer": 7846587,
            "device.firmware.version": "65m",
            "hardware.version.enum": 91,
            "message.buffered.status": true,
            "message.type": "11",
            "position.altitude": 149.19,
            "position.direction": 0,
            "position.hdop": 3,
            "position.latitude": 41.04424,
            "position.longitude": -74.077655,
            "position.satellites": 5,
            "position.speed": 1,
            "private.status": false
 }
```


11.2.2 *Message Type 0: Legacy Tracking*


```
{
            "ain.1": 221,
            "ain.2": 0,
            "ain.3": 146,
            "ain.4": 0,
            "channel.id": 71085,
            "device.id": 2260949,
            "device.io.bitmask.1": 14,
            "device.io.bitmask.2": 0,
            "device.io.bitmask.3": 0,
            "device.io.bitmask.4": 0,
            "device.name": "2460648",
            "device.type.id": 37,
            "external.alarm.status": false,
            "gnss.status": true,
            "gps.mode.1": 64,
            "gps.mode.2": 1,
            "gsm.mcc": "310",
            "gsm.mnc": "410",
            "gsm.network.roaming.status": false,
            "hardware.type": "1B",
            "hardware.version.enum": 91,
            "ident": "2460648",
```

```
            "message.buffered.status": true,
            "message.type": "0",
            "modem.type.enum": 2,
            "operation.mode.enum": "standby engine off",
            "peer": "162.244.250.197:48582",
            "position.direction": 0,
            "position.satellites": 0,
            "position.valid": false,
            "private.status": false,
            "protocol.id": 20,
            "protocol.version": "4",
            "report.reason": 70,
            "report.reason.data": 0,
            "server.timestamp": 1634659336.779554,
            "software.version": "65m",
            "speed.source": "gps",
            "time.valid.status": true,
            "timestamp": 1634659254,
            "vehicle.odometer": 0
 }
```


11.3 CelloTrack 10Y

11.3.1 *Message Type 11 Module 41: Tracking Data*


```
{
            "battery.health": 100,
            "ain.1": 255,
            "ain.2": 100,
            "ain.3": 141,
            "ain.4": 0,
            "channel.id": 71085,
            "device.firmware.version": "65p",
            "device.id": 2362481,
            "device.io.bitmask.1": 32,
            "device.io.bitmask.2": 0,
            "device.io.bitmask.3": 8,
            "device.io.bitmask.4": 0,
            "device.name": "2447540",
            "device.type.id": 799,
            "gnss.status": true,
            "gps.mode.1": 4,
            "gps.mode.2": 2,
            "gsm.network.roaming.status": false,
            "hardware.version.enum": 47,
            "ident": "2447540",
            "message.buffered.status": true,
```

```
            "message.type": "11",
            "module.id": 41,
            "operation.mode.enum": "standby engine off",
            "peer": "162.244.250.197:16901",
            "position.altitude": 134.6,
            "position.direction": 0,
            "position.hdop": 1,
            "position.latitude": 41.075409,
            "position.longitude": -74.163379,
            "position.satellites": 5,
            "position.speed": 1,
            "private.status": false,
            "protocol.id": 20,
            "protocol.version": "4",
            "report.reason": 69,
            "report.reason.data": 0,
            "server.timestamp": 1637337138.367889,
            "speed.source": "gps",
            "time.valid.status": true,
            "timestamp": 1637267783,
            "vehicle.odometer": 0
 }
```


11.3.2 *Message Type 0: Legacy Location*


```
{
            "ain.1": 255,
            "ain.2": 100,
            "ain.3": 137,
            "ain.4": 0,
            "channel.id": 71085,
            "device.id": 2362481,
            "device.io.bitmask.1": 32,
            "device.io.bitmask.2": 0,
            "device.io.bitmask.3": 8,
            "device.io.bitmask.4": 0,
            "device.name": "2447540",
            "device.type.id": 799,
            "external.alarm.status": false,
            "gnss.status": true,
            "gps.mode.1": 0,
            "gps.mode.2": 0,
            "gsm.mcc": "310",
            "gsm.mnc": "410",
            "gsm.network.roaming.status": false,
            "hardware.type": "0F",
            "hardware.version.enum": 47,
```

```
            "ident": "2447540",
            "message.buffered.status": false,
            "message.type": "0",
            "modem.type.enum": 1,
            "operation.mode.enum": "standby engine off",
            "peer": "162.244.250.197:49607",
            "position.altitude": 141.19,
            "position.direction": 26.642538,
            "position.latitude": 41.075187,
            "position.longitude": -74.163362,
            "position.satellites": 7,
            "position.speed": 0.2,
            "private.status": false,
            "protocol.id": 20,
            "protocol.version": "4",
            "report.reason": 31,
            "report.reason.data": 0,
            "server.timestamp": 1637771979.510444,
            "software.version": "65p",
            "speed.source": "gps",
            "time.valid.status": true,
            "timestamp": 1637771978,
            "vehicle.odometer": 0
 }
```


11.4 Nano

11.4.1 *Message Type 11 Module 41: Tracking Data*


```
{
            "external.alarm.status": false,
            "gsm.mcc": "310",
            "gsm.mnc": "410",
            "hardware.type": "1A",
            "modem.type.enum": 3,
            "software.version": "34p",
            "ain.1": 203,
            "ain.2": 216,
            "ain.3": 148,
            "ain.4": 24,
            "channel.id": 71085,
            "device.firmware.version": "34p",
            "device.id": 2211229,
            "device.io.bitmask.1": 68,
            "device.io.bitmask.2": 2,
            "device.io.bitmask.3": 0,
            "device.io.bitmask.4": 128,
```

```
            "device.name": "Nano 3G (TAG MODE)",
            "device.type.id": 34,
            "gnss.status": true,
            "gps.mode.1": 0,
            "gps.mode.2": 0,
            "gsm.network.roaming.status": false,
            "hardware.version.enum": 122,
            "ident": "2338718",
            "message.buffered.status": true,
            "message.type": "11",
            "module.id": 41,
            "operation.mode.enum": "standby engine off",
            "peer": "162.244.250.223:42762",
            "position.direction": 0,
            "position.hdop": 0,
            "position.satellites": 0,
            "position.valid": false,
            "private.status": false,
            "protocol.id": 20,
            "protocol.version": "4",
            "report.reason": 207,
            "report.reason.data": 2,
            "server.timestamp": 1637775565.609996,
            "speed.source": "gps",
            "time.valid.status": true,
            "timestamp": 1637771771,
            "vehicle.odometer": 0,
            "air.pressure.altitude": -71.3,
            "ambient.illumination": 0,
            "battery.level": 0,
            "device.temperature": 24.4,
            "gsm.signal.dbm": -127,
            "temperature.valid": false,
            "x.acceleration": 0.06,
            "y.acceleration": 0.096,
            "z.acceleration": -1.016
 }
```


11.4.2 *Message Type 11 Module 40: Sensor Record*


```
{
            "sensor.temperature": 22.4,
            "channel.id": 71085,
            "device.firmware.version": "34p",
            "device.id": 2211229,
            "device.name": "Nano 3G (TAG MODE)",
```

```
            "device.type.id": 34,
            "gps.mode.1": 4,
            "gps.mode.2": 2,
            "hardware.version.enum": 122,
            "ident": "2338718",
            "measurement.source": 0,
            "message.buffered.status": true,
            "message.type": "11",
            "module.id": 40,
            "peer": "162.244.250.223:42762",
            "position.altitude": 7.8,
            "position.direction": 188.789594,
            "position.hdop": 3,
            "position.latitude": 25.738866,
            "position.longitude": -80.261862,
            "position.satellites": 5,
            "position.speed": 1,
            "private.status": false,
            "protocol.id": 20,
            "sensor.humidity": 52.5,
            "server.timestamp": 1637775569.386037,
            "time.valid.status": true,
            "timestamp": 1637771789
 }
```


11.4.3 *Message Type 0: Legacy Tracking*


```
{
            "ain.1": 216,
            "ain.2": 209,
            "ain.3": 153,
            "ain.4": 24,
            "channel.id": 71085,
            "device.id": 2211229,
            "device.io.bitmask.1": 68,
            "device.io.bitmask.2": 2,
            "device.io.bitmask.3": 136,
            "device.io.bitmask.4": 133,
            "device.name": "Nano 3G (TAG MODE)",
            "device.type.id": 34,
            "external.alarm.status": false,
            "gnss.status": true,
            "gps.mode.1": 0,
            "gps.mode.2": 0,
            "gsm.mcc": "310",
            "gsm.mnc": "410",
            "gsm.network.roaming.status": false,
```

```
            "hardware.type": "1A",
            "hardware.version.enum": 122,
            "ident": "2338718",
            "message.buffered.status": true,
            "message.type": "0",
            "modem.type.enum": 3,
            "operation.mode.enum": "standby engine off",
            "peer": "162.244.250.223:42762",
            "position.direction": 0,
            "position.satellites": 0,
            "position.valid": false,
            "private.status": false,
            "protocol.id": 20,
            "protocol.version": "4",
            "report.reason": 207,
            "report.reason.data": 3,
            "server.timestamp": 1637775568.701029,
            "software.version": "34p",
            "speed.source": "gps",
            "time.valid.status": true,
            "timestamp": 1637772464,
            "vehicle.odometer": 0
 }
```


11.5 MultiSense

11.5.1 *Module Type 28: Event Records*


```
{
            "channel.id": 71085,
            "device.id": 2260949,
            "device.name": "2460648",
            "device.type.id": 37,
            "event.enum": 17,
            "ident": "2460648",
            "module.id": 28,
            "peer": "162.244.250.197:48582",
            "protocol.id": 20,
            "sensor.battery.voltage": 2.937,
            "sensor.mac.address": "481A8400642D",
            "sensor.rssi": -54,
            "sensor.zone": 0,
            "server.timestamp": 1634659262.142391,
            "timestamp": 1634659262.142391
 }
```

11.5.2 *Message Type 11 Module 28 (Tag Mode): Events*


```
{
            "sensor.power.on.status": true,
            "air.pressure.altitude": -71.3,
            "ambient.illumination": 0,
            "battery.level": 0,
            "device.temperature": 24.4,
            "gsm.signal.dbm": -127,
            "temperature.valid": false,
            "x.acceleration": 0.0575,
            "y.acceleration": 0.08175,
            "z.acceleration": -0.98425,
            "package.status": false,
            "sensor.illumination": 12.5,
            "sensor.mac.address": "481A8400452D",
            "sensor.timestamp": 1637771793,
            "package.open.status": false,
            "ble.signal.dbm": -45,
            "box.status": false,
            "channel.id": 71085,
            "device.firmware.version": "34p",
            "device.id": 2211229,
            "device.name": "Nano 3G (TAG MODE)",
            "device.type.id": 34,
            "event.enum": 7,
            "gps.mode.1": 64,
            "gps.mode.2": 1,
            "hardware.version.enum": 122,
            "ident": "2338718",
            "measurement.source": 0,
            "message.buffered.status": true,
            "message.type": "11",
            "module.id": 28,
            "peer": "162.244.250.223:42762",
            "position.direction": 0,
            "position.hdop": 0,
            "position.satellites": 0,
            "position.valid": false,
            "private.status": false,
            "protocol.id": 20,
            "sensor.battery.level": 5,
            "sensor.battery.low.status": true,
            "sensor.humidity": 52.5,
            "sensor.illuminance": 12.5,
            "sensor.magnet.status": false,
            "sensor.temperature": 22.4,
```

```
            "sensor.x.acceleration": -0.192,
            "sensor.y.acceleration": -0.192,
            "sensor.z.acceleration": 0.96,
            "server.timestamp": 1637775567.100436,
            "time.valid.status": true,
            "timestamp": 1637771793
 }
```


11.5.3 *Message Type 11 Module 44: Additional Sensor Data*


```
{
            "channel.id": 71085,
            "device.firmware.version": "34p",
            "device.id": 2211229,
            "device.name": "Nano 3G (TAG MODE)",
            "device.type.id": 34,
            "gps.mode.1": 4,
            "gps.mode.2": 2,
            "hardware.version.enum": 122,
            "ident": "2338718",
            "message.buffered.status": true,
            "message.type": "11",
            "peer": "162.244.250.223:42762",
            "position.altitude": 7.8,
            "position.direction": 188.789594,
            "position.hdop": 3,
            "position.latitude": 25.738866,
            "position.longitude": -80.261862,
            "position.satellites": 5,
            "position.speed": 1,
            "private.status": false,
            "protocol.id": 20,
            "server.timestamp": 1637775569.882174,
            "time.valid.status": true,
            "timestamp": 1637772678,
            "ble.signal.dbm": -53,
            "module.id": 44,
            "package.status": false,
            "sensor.battery.level": 100,
            "sensor.humidity": 6553.5,
            "sensor.illumination": 7,
            "sensor.mac.address": "481A84002407",
            "sensor.magnet.status": false,
            "sensor.temperature": 22.8,
            "sensor.timestamp": 1637772597,
            "sensor.x.acceleration": -0.192,
            "sensor.y.acceleration": 0.064,
```

```
            "sensor.z.acceleration": -0.96,
            "air.pressure.altitude": -71.4,
            "ambient.illumination": 0,
            "battery.level": 24,
            "device.temperature": 24.4,
            "gsm.signal.dbm": -89,
            "temperature.valid": false,
            "x.acceleration": 0.0075,
            "y.acceleration": -0.0065,
            "z.acceleration": -1.01625
}
```