Myriota

Myriota AssetHawk™

Message Format Description

XAN1110-02

Apr 2026

Revision History

Rev	Date	Description of Change
01	Jan 2026	Initial version
02	Apr 2026	Added Tamper message reason Added Message Flag Detail Added BLE message format

Related Documentation

Find the latest versions of all Myriota documentation at developer.myriota.com

How to Contact Us

Technical Support

developer.myriota.com

support.myriota.com

Sales Support

sales@myriota.com

Myriota Online

myriota.com

Disclaimer

The information contained in this document (collectively, the “Information”) is provided to you (both the individual receiving this document and any legal entity on behalf of which such individual is acting) (“You” and “Your”) by Myriota Pty Ltd for information purposes only.

- Information in this document is provided solely to enable system and software implementers to use Myriota Pty Ltd products.
- Myriota Pty Ltd reserves the right to make changes without further notice to any products herein.
- You are responsible for making Your own assessments concerning the Information and Myriota recommends that You assess the accuracy, completeness and relevance of the Information for Your purposes before using or relying on any of the Information.
- Myriota is providing the Information to you “AS IS” and without regard to Your specific requirements.
- Myriota has exercised reasonable care in preparing the Information, however Myriota does not warrant the accuracy, completeness or relevance of the Information and accepts no liability for any errors or omissions in the Information.
- You acknowledge and agree that Your use of the Information is at your sole risk and that to the extent permitted by law Myriota is not liable for any loss or damage of whatever nature (direct, indirect, consequential or other) that arises in any way from Your use of or reliance on the Information.
- For further information, see myriota.com or contact your Myriota sales representative.

Table of Contents

Overview

This document details the data structure of the messages from AssetHawk™ devices, which will be sent via HTTPS POST requests to customer visualisation platforms. The data received by the platform is in standard JSON format.

The payload contains telemetry data from the AssetHawk™, including location, device health (battery and temperature), and the specific reason the report was triggered (e.g., movement detected, stationary timeout, etc.).

Note that messages may be duplicated and/or arrive out of order.

POST request structure

The body of the POST request is formatted as JSON. The example below illustrates the structure of the request.

```
{
  "EndpointRef": "N_HlfTNgRsqh:uyXKvYTmTAO5.",
  "Timestamp": 1563521870,
  "Data":
        "{\"Packets\":
              [{\"Timestamp\": 1563521870359,
              \"TerminalId\": \"0001020304\",
              \"Value\": \"MessagePayload\"}]
        }",
  "Id": "fe77e2c7-8e9c-40d0-8980-43720b9dab75",
  "CertificateUrl":    "https://security.myriota.com/data-13f71.crt",
  "Signature": "k2OlBppBXmBT520rUllvMxNg+h9soJYBhQhOGSlWGdzkppdT1.."
}
```

The table below provides a breakdown of the corresponding fields in the POST request.

<table>
<tr><th>Field</th><th>Type</th><th>Description</th></tr>
<tr><td>EndpointRef</td><td>String</td><td>Unique Identifier of the destination</td></tr>
<tr><td>Timestamp</td><td>Number</td><td>Unix Epoch time at which the HTTP POST request was generated</td></tr>
<tr><td>Data</td><td>Object</td><td>A JSON serialised structure with the following fields:
<table>
<tr><td>Packets</td><td>List</td><td>JSON list of the packets received from the device. For AssetHawk, this list only contains one message.</td></tr>
<tr><td>Timestamp</td><td>Number</td><td>Unix Epoch time (ms) at which the packet was captured by the satellite</td></tr>
<tr><td>TerminalId</td><td>Number</td><td>Serial number of the AssetHawk device from which the packet was transmitted</td></tr>
<tr><td>Value</td><td>Object</td><td>The payload data from the device. See Payload Format for more information.</td></tr>
</table></td></tr>
<tr><td>Id</td><td>String</td><td>HTTPS Message UUID</td></tr>
<tr><td>CertificateUrl</td><td>URL</td><td>URL of certificate containing the public key to authenticate the Signature.</td></tr>
<tr><td>Signature</td><td>String</td><td>PKCS1_15 signature of Timestamp, Id and Data fields</td></tr>
</table>

Details on how to use the Certificate and Signature fields to validate the data are outlined on the Myriota Support site here.

Payload Format

Location Message

The payload data is in JSON format, as shown in the example location payload data below:


```
"Value": {
        "type": 1,
        "sequence_number": 10,
        "reason": 0,
        "flags": 2,
        "time": 1738183958,
        "latitude": -34.9275233,
        "longitude": 138.6075245,
         "elevation": 63.032,
         "temperature": 18,
         "battery_voltage": 3.600
}
```


Location Payload Breakdown

The table below describes each field in the JSON location payload and its unit of measurement (UOM).

<table>
<tr><th>Content</th><th>UOM</th><th>Note</th></tr>
<tr><td>Message Type</td><td>N/A</td><td>Message versioning tracks changes to the message structure.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>Location Message</td></tr>
<tr><td>8</td><td>Bluetooth Message</td></tr>
</table>
</td></tr>
<tr><td>Sequence Number</td><td>N/A</td><td>An incrementing number to keep track of the message sequence, i.e. message count. The</td></tr>
</table>

<table>
<tr><td></td><td></td><td>sequence number has a maximum value of 255; after that, it wraps around to 0 and continues counting. If this number resets to 0 before reaching 255, the device has reset/rebooted.

This number is increased separately for each Message Type.</td></tr>
<tr><td>Message Reason</td><td></td><td>The reason the message was sent. The reason is specified as a number (enumeration). The table below defines each value.

<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>The first message sent after boot. Triggered ~60 seconds after startup.</td></tr><tr><td>1</td><td>Stationary interval elapsed: the device has been stationary for the configured stationary interval.</td></tr><tr><td>2</td><td>Movement started: the device was stationary, and movement has now been detected.</td></tr><tr><td>3</td><td>Movement stopped: the device was in motion and has since stopped.</td></tr><tr><td>4</td><td>Movement interval elapsed: The device was in motion and remains in motion.</td></tr><tr><td>5</td><td>Reserved for future use.</td></tr><tr><td>6</td><td>The device has been tampered with.</td></tr></table></td></tr>
<tr><td>Message Flag</td><td>N/A</td><td>The message flag highlights information that is relevant but not related to the reason a message was sent.

<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>No Active flag</td></tr></table></td></tr>
</table>

		1 | Incorrect Orientation: The device is misoriented, which may affect the line of sight to the satellites.
Time	N/A	The time of the location reading, in UTC. This is in Unix time format (also called Epoch time). e.g. 1758520800 is 2025-09-22 06:00:00
Latitude	degrees	The device's latitude in degrees.
Longitude	degrees	The device's longitude in degrees.
Elevation	m	The device's elevation above average sea level, in meters.
Temperature	°C	Internal temperature of the device in degrees Celsius
Battery Voltage	V	The device's battery level in Volts.

Note: *that future firmware releases may add additional fields, reasons and/or flags to the payload data structure*

Bluetooth Message

The payload data is in JSON format, as shown in the example BLE payload data below:

```
"Value": {
        "type": 8,
        "sequence_number": 10,
        "time": 1738183958,
        "protocol": 1,
        "RSSI": -70,
        "address":”AA:BB:CC:DD:EE:FF”,
        "data": “Beacon Data”
}
```

The table below describes each field in the JSON BLE payload and its unit of measurement (UOM).

<table>
<tr><th>Content</th><th>UOM</th><th>Note</th></tr>
<tr><td>Message Type</td><td>N/A</td><td>Message versioning tracks changes to the message structure.<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>Location Message</td></tr><tr><td>8</td><td>Bluetooth Message</td></tr></table></td></tr>
<tr><td>Sequence Number</td><td>N/A</td><td>An incrementing number to keep track of the message sequence, i.e. message count. The sequence number has a maximum value of 255; after that, it wraps around to 0 and continues counting. If this number resets to 0 before reaching 255, the device has reset/rebooted.

This number is increased separately for each Message Type.</td></tr>
<tr><td>Time</td><td>N/A</td><td>The time of the location reading, in UTC. This is in Unix time format (also called Epoch time). e.g. 1758520800 is 2025-09-22 06:00:00</td></tr>
<tr><td>Protocol</td><td>N/A</td><td>BLE advertising protocol type from the beacon.<table><tr><th>Value</th><th>Description</th></tr><tr><td>1</td><td>EddyStone format</td></tr><tr><td>2</td><td>Reserved for future use</td></tr><tr><td>3</td><td>Reserved for future use</td></tr><tr><td>4</td><td>Advertising data or scan response frame</td></tr></table></td></tr>
<tr><td>RSSI</td><td>dBm</td><td>Signal strength between the AssetHawk and the BLE beacon in dBm.</td></tr>
</table>

Address	N/A	MAC address of the beacon.
Data	N/A	Raw beacon payload. The beacon payload is not decoded and is sent as captured from the beacon.

Note: *that future firmware releases may add additional fields, reasons and/or flags to the payload data structure*