Veriflite Portal 0.1-9-gbf97ead

INFO@VERIFLITE.COM ID: HTTPS://VERIFLITE.GITHUB.IO/PORTAL-API/

The Veriflite Portal allows developers direct access to Veriflite's Time of Flight data, to create custom applications on top of the Veriflite platform.

The Portal uses websockets for communication and has three endpoints available: /raw, /flight, and /sensor. The /raw endpoint publishes unprocessed bounce data, while the /flight endpoint publishes processed data. The /sensor endpoint responds to requests for information about the active sensors.

To get started you can use a tool such as websocat, available at https://github.com/vi/websocat, to connect to the endpoints. For example to observe raw bounce data, start the Veriflite app and go to Labs -> Portal; then on your PC, substituting for the IP address displayed in the Portal, run the following command and you should see a stream of bounce events depending on the activity on your trampoline:

```
websocat ws://192.168.1.95:4651/raw
```

There are also some simple example Python projects available at https://github.com/Veriflite/Portal-Demo that demonstrate how to receive Veriflite data.

Please let us know if you're trying the Portal out, we are keen to hear how you get on!

Servers

`ws://{ip}:4651` WS DEVICE

A websocket connection must be opened to the URI displayed by the Veriflite app.

URL Variables ^ Expand all

ip required	**String** The IP address of the device running the Veriflite app.

Operations

SUB `/raw`

Open a connection to the /raw endpoint to receive unprocessed bounce events in the form of landing and depart timestamps.

Accepts the following message:

packet

APPLICATION/JSON

Minimally processed live data from the Veriflite sensors as received by the app.

Payload ^ Expand all **Object** uid: packet

event **String**
packet
Const: "packet"

data ^ Expand all **Object** uid: packetData
Examples values:
{"address":"90:FD:9F:AC:9B:7F","sequenceNumber":11,"type":"IMPACT_TIME","data":9072}

address **String**
Unique sensor identifier.

sequenceNumber **Integer**
Used to determine the order in which the packets were generated by the sensor.

type **String**
Sensor event described in this packet. The _SYNC suffix indicates that the sensor has synchronised its clock with another sensor, and the timestamp values may be used for calculating a synchro score, whereas the _TIME suffix indicates that it is not a synchronised value.
Allowed values: "IMPACT_SYNC" "IMPACT_TIME" "DEPART_SYNC" "DEPART_TIME" "IDLE_SYNC" "IDLE_TIME" "BATTERY_VOLTAGE"

data **Integer**
Sensor packet payload, either a timestamp (ms) or battery voltage (mV).

Additional properties are allowed.

Additional properties are allowed.

Examples

Payload ^

```
{
  "event": "packet",
  "data": {
    "address": "90:FD:9F:AC:9B:7F",
    "sequenceNumber": 11,
    "type": "IMPACT_TIME",
    "data": 9072
  }
}
```

This example has been generated automatically.

SUB /flight

Processed bounce events.

Receive processed bounce data from the Veriflite sensors as it arrives at the app.

Accepts **one of** the following messages:

#0 bounce

APPLICATION/JSON

Live individual bounce data after processing in the Veriflite app. This event may be a new bounce, an update to an already received bounce or indicate an existing bounce has been found invalid and should be discarded.

Payload ^ Expand all **Object** uid: bounce

event **String**
bounce
Const: "bounce"

data ^ Expand all **Object** uid: bounceData
Examples values:
{"address":"90:FD:9F:AC:9B:7F","sequenceNumber":11,"timeOfFlight":1425,"toFDelta":45,"impactTimestamp":9072,"isInvalid":false}

address **String**
Unique sensor identifier.

sequenceNumber **Integer**
The order of the bounce event relative to other events from the same sensor. This value corresponds to the sequence number of the IMPACT packet which ends the flight phase of the bounce. Note - This sequence will not be contiguous.

timeOfFlight **Integer**
Duration (ms) from DEPART to IMPACT event timestamps.

toFDelta **Integer**
Difference (ms) in timeOfFlight from the previous bounce.

impactTimestamp **Integer**
Timestamp (ms) of the IMPACT event which ends the flight phase.

isInvalid **Boolean**
Indicates that a bounceData packet which was previously broadcast is now determined to be invalid and should be discarded.

Additional properties are allowed.

Additional properties are allowed.

Examples

Payload ^

```
{
  "event": "bounce",
  "data": {
    "address": "90:FD:9F:AC:9B:7F",
    "sequenceNumber": 11,
    "timeOfFlight": 1425,
    "toFDelta": 45,
    "impactTimestamp": 9072,
    "isInvalid": false
  }
}
```

This example has been generated automatically.

#1 idle

Event signalling that a Veriflite sensor has gone idle.

APPLICATION/JSON

Payload ^ Expand all **Object** uid: idle

event **String**
idle
Const: "idle"

data ^ Expand all **Object** uid: idleData
Examples values:
{"address":"90:FD:9F:AC:9B:7F","sequenceNumber":12,"impactTimestamp":10352}

address **String**
Unique sensor identifier.

sequenceNumber **Integer**
The order of the idle event relative to other events from the same sensor.

idleTimestamp **Integer**
Timestamp (ms) of the IDLE event.

Additional properties are allowed.

Additional properties are allowed.

Examples

Payload ^

```
{
  "event": "idle",
  "data": {
    "address": "90:FD:9F:AC:9B:7F",
    "sequenceNumber": 12,
    "impactTimestamp": 10352
  }
}
```

This example has been generated automatically.

#2 sensor-reset

Event indicating that at least one of the Veriflite sensors connected to the application has disconnected or reset. Expect sequence numbers following this to reset and caches to be discarded (there will be no updates to past jumps on missing data received).

Payload ^ Expand all **Object** uid: sensor-reset

event **String**
sensor-reset
Const: "sensor-reset"

data ^ Expand all **Object**
Placeholder for future development
Const: null

Additional properties are allowed.

Additional properties are allowed.

Examples

Payload ^

```
{
  "event": "sensor-reset",
  "data": null
}
```

This example has been generated automatically.

#3 missing-data

APPLICATION/JSON

Event indicating that there are packets that were expected from the app that have either not arrived - meaning that jump data is unreliable, or that previously missing packets have arrived.

Payload ^ Expand all **Object** uid: missing-data

event **String**

The app has detected it is missing some data from the sensors.

Const: "missing-data"

data ^ Expand all **Object** uid: missingData

Examples values:

{"address":"90:FD:9F:AC:9B:7F","missingSequenceNumbes":[8,10]}

address **String**

Unique sensor identifier.

missingSequenceNumbes ^ Expand all **Array<Any>**

List of currently missing sequence numbers. If this is empty it means that all previously missing packets have been found.

Additional properties are allowed.

Additional properties are allowed.

Examples

Payload ^

```
{
  "event": "missing-data",
  "data": {
    "address": "90:FD:9F:AC:9B:7F",
    "missingSequenceNumbes": [
      8,
      10
    ]
  }
}
```

This example has been generated automatically.

PUB /sensor

Request sensor data and receive sensor events.

Request information about the Veriflite sensors known by the app.

Accepts **one of** the following messages:

#0 sensor-request

APPLICATION/JSON

Request a list of the sensors known to the app. This consists of sensors that are either currently active or have been active since the Portal was started on the app.

This query will prompt a sensor-list response.

Payload ^ Expand all **Object** uid: sensor-request

request	**String** sensor-list Const: "sensor-list"

Additional properties are allowed.

Examples

Payload ^

```
{
  "request": "sensor-list"
}
```

This example has been generated automatically.

#1 sensor-data

APPLICATION/JSON

Request a summary of what the application knows about a specific sensor.

This query will prompt a sensor-details response.

Payload ^ Expand all **Object** uid: sensor-data

Examples values: {"request":"sensor-data","args":"90:FD:9F:AC:9B:7F"}

request **String**

sensor-details

Const: "sensor-details"

args **String**

The address of a known sensor. Use the sensor-list request to get a list of valid options.

Additional properties are allowed.

Examples

Payload ^

#1 Example

```
{
  "request": "sensor-data",
  "args": "90:FD:9F:AC:9B:7F"
}
```

SUB /sensor

Request sensor data and receive sensor events.

Receive information about the Veriflite sensors known by the app.

Accepts **one of** the following messages:

#0 error

APPLICATION/JSON

Event exposing errors with the socket connection, requests or responses.

Payload ^ Expand all **Object** uid: error

event **String**

error

Const: "error"

data ^ Expand all **Object** uid: errorData

Examples values:

{"error":"InvalidArgument","message":"Could not find a sensor with the address \"90:FD:9F:AC:9B:7F\", request sensor-list for known sensors."}

error **String**
Type of error

message **String**
Short description of the error

Additional properties are allowed.

Additional properties are allowed.

Examples

Payload ^

```
{
  "event": "error",
  "data": {
    "error": "InvalidArgument",
    "message": "Could not find a sensor with the address \"90:FD:9F:AC:9B:7F\", request sensor-list for known sensors."
  }
}
```

This example has been generated automatically.

#1 sensor-list

APPLICATION/JSON

A list of the sensors known to the app. This consists of sensors that are either currently active or have been active since the Portal was started on the app.

This is a response to the sensor-list query.

Payload ^ Expand all **Object** uid: sensor-list

event **String**
sensor-list
Const: "sensor-list"

data ^ Expand all **Object** uid: sensorListData
Examples values: {"sensors":["90:FD:9F:AC:9B:7F","CC:CC:CC:9B:D8:FD"]}

sensors ^ Expand all **Array<Any>**
List of sensors that have been seen by the app while the Portal has been active.

Additional properties are allowed.

Additional properties are allowed.

Examples

Payload ^

```
{
  "event": "sensor-list",
  "data": {
    "sensors": [
      "90:FD:9F:AC:9B:7F",
      "CC:CC:CC:9B:D8:FD"
    ]
  }
}
```

This example has been generated automatically.

#2 sensor-details

APPLICATION/JSON

A summary of what the app knows about a specific sensor.

This is a response to the sensor-list query.

Payload ^ Expand all **Object** uid: sensor-details

event **String**

sensor-details

Const: "sensor-details"

data ^ Expand all **Object** uid: sensorDetailData

Examples values:

{"friendlyName":"Trampoline 1","firmwareVersion":"1.13.3","batteryLevel":1503,"isPaired":false,"syncID":-1,"bootNumber":173,"lifetimeJumps":4542,"lifetimeAge":188678,"uniqueID":844383776,"advertisingRate":13.0791388,"isHighPower":false}

friendlyName **String**

User friendly name given to the sensor and displayed throughout Veriflite

firmwareVersion **String**

Firmware version running on the sensor

batteryLevel **Integer**

Battery voltage in mV

isPaired **Boolean**

Whether the sensor is paired (and clock synced) with another sensor. Useful for synchro calculations.

syncID

Integer

Given to a sensor for synchro pairing. Sensors will attempt to pair with others on the same sync ID.

bootNumber

Integer

Number of times the sensor has been booted up in its lifetime.

lifetimeJumps

Integer

Total number of bounce events sent by the sensor in its lifetime.

lifetimeAge

Integer

Total time the sensor has been active in its lifetime, measured in seconds.

uniqueID

Integer

A unique ID number for a given sensor.

advertisingRate

Number

Rate at which packets from the sensor are received by the app - indicates the strength of the signal from the sensor to app.

isHighPower

Boolean

Whether the sensor is in high power mode (sending out packets at a higher rate).

Additional properties are allowed.

Additional properties are allowed.

Examples

Payload

```
{
  "event": "sensor-details",
  "data": {
    "friendlyName": "Trampoline 1",
    "firmwareVersion": "1.13.3",
    "batteryLevel": 1503,
    "isPaired": false,
    "syncID": -1,
    "bootNumber": 173,
    "lifetimeJumps": 4542,
    "lifetimeAge": 188678,
    "uniqueID": 844383776,
    "advertisingRate": 13.0791388,
    "isHighPower": false
  }
}
```

This example has been generated automatically.

#3 battery

Current sensor battery level (mV)

APPLICATION/JSON

Payload ^ Expand all **Object** uid: battery

event **String**
battery
Const: "battery"

data ^ Expand all **Object** uid: batteryData
Examples values: {"address":"90:FD:9F:AC:9B:7F","batteryLevel":1503}

address **String**
Unique sensor identifier.

batteryLevel **Integer**
Battery voltage in mV

Additional properties are allowed.

Additional properties are allowed.

Examples

Payload ^

```
{
  "event": "battery",
  "data": {
    "address": "90:FD:9F:AC:9B:7F",
    "batteryLevel": 1503
  }
}
```

This example has been generated automatically.

#4 status-flags

APPLICATION/JSON

Status flags received from a sensor providing firmware information.

Payload ^ Expand all **Object** uid: status-flags

event **String**

status-flags

Const: "status-flags"

data ^ Expand all **Object** uid: statusFlagData

Examples values:

{"address":"90:FD:9F:AC:9B:7F","uniqueID":844383776,"firmwareVersion":"1.13.3","isHighPower":false,"bootNumber":134,"lifetimeJumps":4542,"lifetimeAge":188678}

address **String**

Unique sensor identifier.

uniqueID **Integer**

A unique ID number for a given sensor.

firmwareVersion **String**

Firmware version running on the sensor.

isHighPower **Boolean**

Whether the sensor is in high power mode (sending out packets at a higher rate).

bootNumber **Integer**

Number of times the sensor has been booted up in its lifetime.

lifetimeJumps **Integer**

Total number of bounce events ever detected by the sensor.

lifetimeAge **Integer**

Total time the sensor has been active in its lifetime, measured in seconds.

Additional properties are allowed.

Additional properties are allowed.

Examples

Payload ^

```
{
  "event": "status-flags",
  "data": {
    "address": "90:FD:9F:AC:9B:7F",
    "uniqueID": 844383776,
    "firmwareVersion": "1.13.3",
    "isHighPower": false,
    "bootNumber": 134,
    "lifetimeJumps": 4542,
    "lifetimeAge": 188678
  }
}
```

This example has been generated automatically.

#5 display-name

APPLICATION/JSON

Event indicating that a sensor has sent the user-friendly display name given to it.

Payload ^ Expand all **Object** uid: display-name

event **String**
display-name
Const: "display-name"

data ^ Expand all **Object** uid: displayNameData
Examples values: {"address":"90:FD:9F:AC:9B:7F","friendlyName":"Trampoline 1"}

address **String**
Unique sensor identifier.

friendlyName **String**
User friendly name given to the sensor and displayed throughout Veriflite.

Additional properties are allowed.

Additional properties are allowed.

Examples

Payload ^

```
{
  "event": "display-name",
  "data": {
    "address": "90:FD:9F:AC:9B:7F",
    "friendlyName": "Trampoline 1"
  }
}
```

This example has been generated automatically.

Messages

#1 error

APPLICATION/JSON

Event exposing errors with the socket connection, requests or responses.

Payload ^ Expand all **Object** uid: error

event **String**

error

Const: "error"

data ^ Expand all **Object** uid: errorData

Examples values:

```
{"error":"InvalidArgument","message":"Could not find a sensor with the address \"90:FD:9F:AC:9B:7F\", request sensor-list for known sensors."}
```

error **String**

Type of error

message **String**

Short description of the error

Additional properties are allowed.

Additional properties are allowed.

#2 packet

APPLICATION/JSON

Minimally processed live data from the Veriflite sensors as received by the app.

Payload ^ Expand all **Object** uid: packet

event **String**

packet

Const: "packet"

data ^ Expand all **Object** uid: packetData

Examples values:

```
{"address":"90:FD:9F:AC:9B:7F","sequenceNumber":11,"type":"IMPACT_TIME","data":9072}
```

address **String**

Unique sensor identifier.

sequenceNumber **Integer**

Used to determine the order in which the packets were generated by the sensor.

type

String

Sensor event described in this packet. The _SYNC suffix indicates that the sensor has synchronised its clock with another sensor, and the timestamp values may be used for calculating a synchro score, whereas the _TIME suffix indicates that it is not a synchronised value.

Allowed values: "IMPACT_SYNC" "IMPACT_TIME" "DEPART_SYNC" "DEPART_TIME" "IDLE_SYNC" "IDLE_TIME" "BATTERY_VOLTAGE"

data

Integer

Sensor packet payload, either a timestamp (ms) or battery voltage (mV).

Additional properties are allowed.

Additional properties are allowed.

#3 bounce

APPLICATION/JSON

Live individual bounce data after processing in the Veriflite app. This event may be a new bounce, an update to an already received bounce or indicate an existing bounce has been found invalid and should be discarded.

Payload ^ Expand all **Object** uid: bounce

event

String

bounce

Const: "bounce"

data ^ Expand all **Object** uid: bounceData

Examples values:

{"address":"90:FD:9F:AC:9B:7F","sequenceNumber":11,"timeOfFlight":1425,"toFDelta":45,"impactTimestamp":9072,"isInvalid":false}

address

String

Unique sensor identifier.

sequenceNumber

Integer

The order of the bounce event relative to other events from the same sensor. This value corresponds to the sequence number of the IMPACT packet which ends the flight phase of the bounce. Note - This sequence will not be contiguous.

timeOfFlight

Integer

Duration (ms) from DEPART to IMPACT event timestamps.

toFDelta **Integer**
Difference (ms) in timeOfFlight from the previous bounce.

impactTimestamp **Integer**
Timestamp (ms) of the IMPACT event which ends the flight phase.

isInvalid **Boolean**
Indicates that a bounceData packet which was previously broadcast is now determined to be invalid and should be discarded.

Additional properties are allowed.

Additional properties are allowed.

#4 idle

Event signalling that a Veriflite sensor has gone idle.

APPLICATION/JSON

Payload ^ Expand all **Object** uid: idle

event **String**
idle
Const: "idle"

data ^ Expand all **Object** uid: idleData
Examples values:
`{"address":"90:FD:9F:AC:9B:7F","sequenceNumber":12,"impactTimestamp":10352}`

address **String**
Unique sensor identifier.

sequenceNumber **Integer**
The order of the idle event relative to other events from the same sensor.

idleTimestamp **Integer**
Timestamp (ms) of the IDLE event.

Additional properties are allowed.

Additional properties are allowed.

#5 sensor-reset

Event indicating that at least one of the Veriflite sensors connected to the application has disconnected or reset. Expect sequence numbers following this to reset and caches to be discarded (there will be no updates to past jumps on missing data received).

Payload ^ Expand all **Object** uid: sensor-reset

event **String**

sensor-reset

Const: "sensor-reset"

data ^ Expand all **Object**

Placeholder for future development

Const: null

Additional properties are allowed.

Additional properties are allowed.

#6 missing-data

APPLICATION/JSON

Event indicating that there are packets that were expected from the app that have either not arrived - meaning that jump data is unreliable, or that previously missing packets have arrived.

Payload ^ Expand all **Object** uid: missing-data

event **String**

The app has detected it is missing some data from the sensors.

Const: "missing-data"

data ^ Expand all **Object** uid: missingData

Examples values:

{"address":"90:FD:9F:AC:9B:7F","missingSequenceNumbes":[8,10]}

address **String**

Unique sensor identifier.

missingSequenceNumbes ^ Expand all **Array<Any>**

List of currently missing sequence numbers. If this is empty it means that all previously missing packets have been found.

Additional properties are allowed.

Additional properties are allowed.

#7 sensor-request

APPLICATION/JSON

Request a list of the sensors known to the app. This consists of sensors that are either currently active or have been active since the Portal was started on the app.

This query will prompt a sensor-list response.

Payload ^ Expand all **Object** uid: sensor-request

request **String**
sensor-list
Const: "sensor-list"

Additional properties are allowed.

#8 sensor-data

APPLICATION/JSON

Request a summary of what the application knows about a specific sensor.

This query will prompt a sensor-details response.

Payload ^ Expand all **Object** uid: sensor-data
Examples values: {"request":"sensor-data","args":"90:FD:9F:AC:9B:7F"}

request **String**
sensor-details
Const: "sensor-details"

args **String**
The address of a known sensor. Use the sensor-list request to get a list of valid options.

Additional properties are allowed.

#9 sensor-list

APPLICATION/JSON

A list of the sensors known to the app. This consists of sensors that are either currently active or have been active since the Portal was started on the app.

This is a response to the sensor-list query.

Payload ^ Expand all **Object** uid: sensor-list

event **String**
sensor-list
Const: "sensor-list"

data ^ Expand all **Object** uid: sensorListData
Examples values: {"sensors":["90:FD:9F:AC:9B:7F","CC:CC:CC:9B:D8:FD"]}

sensors ^ Expand all

Array<Any>

List of sensors that have been seen by the app while the Portal has been active.

Additional properties are allowed.

Additional properties are allowed.

#10 sensor-details

APPLICATION/JSON

A summary of what the app knows about a specific sensor.

This is a response to the sensor-list query.

Payload ^ Expand all **Object** uid: sensor-details

event

String

sensor-details

Const: "sensor-details"

data ^ Expand all **Object** uid: sensorDetailData

Examples values:

{"friendlyName":"Trampoline 1","firmwareVersion":"1.13.3","batteryLevel":1503,"isPaired":false,"syncID":-1,"bootNumber":173,"lifetimeJumps":4542,"lifetimeAge":188678,"uniqueID":844383776,"advertisingRate":13.0791388,"isHighPower":false}

friendlyName

String

User friendly name given to the sensor and displayed throughout Veriflite

firmwareVersion

String

Firmware version running on the sensor

batteryLevel

Integer

Battery voltage in mV

isPaired

Boolean

Whether the sensor is paired (and clock synced) with another sensor. Useful for synchro calculations.

syncID

Integer

Given to a sensor for synchro pairing. Sensors will attempt to pair with others on the same sync ID.

bootNumber

Integer

Number of times the sensor has been booted up in its lifetime.

Integer

lifetimeJumps

Total number of bounce events sent by the sensor in its lifetime.

lifetimeAge

Integer

Total time the sensor has been active in its lifetime, measured in seconds.

uniqueID

Integer

A unique ID number for a given sensor.

advertisingRate

Number

Rate at which packets from the sensor are received by the app - indicates the strength of the signal from the sensor to app.

isHighPower

Boolean

Whether the sensor is in high power mode (sending out packets at a higher rate).

Additional properties are allowed.

Additional properties are allowed.

#11 battery

Current sensor battery level (mV)

APPLICATION/JSON

Payload ^ Expand all **Object** uid: battery

event

String

battery

Const: "battery"

data ^ Expand all

Object uid: batteryData

Examples values: {"address":"90:FD:9F:AC:9B:7F","batteryLevel":1503}

address

String

Unique sensor identifier.

batteryLevel

Integer

Battery voltage in mV

Additional properties are allowed.

Additional properties are allowed.

#12 status-flags

APPLICATION/JSON

Status flags received from a sensor providing firmware information.

Payload ^ Expand all **Object** uid: status-flags

event **String**
status-flags
Const: "status-flags"

data ^ Expand all **Object** uid: statusFlagData
Examples values:

```
{"address":"90:FD:9F:AC:9B:7F","uniqueID":844383776,"firmwareVersion":"1.13.3","isHighPower":false,"bootNumber":134,"lifetimeJumps":4542,"lifetimeAge":188678}
```

address **String**
Unique sensor identifier.

uniqueID **Integer**
A unique ID number for a given sensor.

firmwareVersion **String**
Firmware version running on the sensor.

isHighPower **Boolean**
Whether the sensor is in high power mode (sending out packets at a higher rate).

bootNumber **Integer**
Number of times the sensor has been booted up in its lifetime.

lifetimeJumps **Integer**
Total number of bounce events ever detected by the sensor.

lifetimeAge **Integer**
Total time the sensor has been active in its lifetime, measured in seconds.

Additional properties are allowed.

Additional properties are allowed.

#13 display-name

APPLICATION/JSON

Event indicating that a sensor has sent the user-friendly display name given to it.

Payload ^ Expand all **Object** uid: display-name

event **String**
display-name

Const: "display-name"

data ^ Expand all **Object** uid: displayNameData

Examples values: {"address":"90:FD:9F:AC:9B:7F","friendlyName":"Trampoline 1"}

address **String**

Unique sensor identifier.

friendlyName **String**

User friendly name given to the sensor and displayed throughout Veriflite.

Additional properties are allowed.

Additional properties are allowed.

Schemas

error ^ Expand all **Object** uid: error

event **String**

error

Const: "error"

data ^ Expand all **Object** uid: errorData

Examples values:

{"error":"InvalidArgument","message":"Could not find a sensor with the address \"90:FD:9F:AC:9B:7F\", request sensor-list for known sensors."}

error **String**

Type of error

message **String**

Short description of the error

Additional properties are allowed.

Additional properties are allowed.

errorData ^ Expand all **Object** uid: errorData

Examples values:

{"error":"InvalidArgument","message":"Could not find a sensor with the address \"90:FD:9F:AC:9B:7F\", request sensor-list for known sensors."}

error **String**

Type of error

message **String**
Short description of the error

Additional properties are allowed.

packet ^ Expand all **Object** uid: packet

event **String**
packet
Const: "packet"

data ^ Expand all **Object** uid: packetData
Examples values:
{"address":"90:FD:9F:AC:9B:7F","sequenceNumber":11,"type":"IMPACT_TIME","data":9072}

address **String**
Unique sensor identifier.

sequenceNumber **Integer**
Used to determine the order in which the packets were generated by the sensor.

type **String**
Sensor event described in this packet. The _SYNC suffix indicates that the sensor has synchronised its clock with another sensor, and the timestamp values may be used for calculating a synchro score, whereas the _TIME suffix indicates that it is not a synchronised value.
Allowed values: "IMPACT_SYNC" "IMPACT_TIME" "DEPART_SYNC" "DEPART_TIME" "IDLE_SYNC" "IDLE_TIME" "BATTERY_VOLTAGE"

data **Integer**
Sensor packet payload, either a timestamp (ms) or battery voltage (mV).

Additional properties are allowed.

Additional properties are allowed.

packetData ^ Expand all **Object** uid: packetData
Examples values:
{"address":"90:FD:9F:AC:9B:7F","sequenceNumber":11,"type":"IMPACT_TIME","data":9072}

address **String**
Unique sensor identifier.

sequenceNumber **Integer**

Used to determine the order in which the packets were generated by the sensor.

type **String**

Sensor event described in this packet. The _SYNC suffix indicates that the sensor has synchronised its clock with another sensor, and the timestamp values may be used for calculating a synchro score, whereas the _TIME suffix indicates that it is not a synchronised value.

Allowed values: "IMPACT_SYNC" "IMPACT_TIME" "DEPART_SYNC" "DEPART_TIME" "IDLE_SYNC" "IDLE_TIME" "BATTERY_VOLTAGE"

data **Integer**

Sensor packet payload, either a timestamp (ms) or battery voltage (mV).

Additional properties are allowed.

bounce ^ Expand all **Object** uid: bounce

event **String**

bounce

Const: "bounce"

data ^ Expand all **Object** uid: bounceData

Examples values:

```
{"address":"90:FD:9F:AC:9B:7F","sequenceNumber":11,"timeOfFlight":1425,"toFDelta":45,"impactTimestamp":9072,"isInvalid":false}
```

address **String**

Unique sensor identifier.

sequenceNumber **Integer**

The order of the bounce event relative to other events from the same sensor. This value corresponds to the sequence number of the IMPACT packet which ends the flight phase of the bounce. Note - This sequence will not be contiguous.

timeOfFlight **Integer**

Duration (ms) from DEPART to IMPACT event timestamps.

toFDelta **Integer**

Difference (ms) in timeOfFlight from the previous bounce.

impactTimestamp **Integer**

Timestamp (ms) of the IMPACT event which ends the flight phase.

isInvalid **Boolean**

Indicates that a bounceData packet which was previously broadcast is now determined to be invalid and should be discarded.

Additional properties are allowed.

Additional properties are allowed.

bounceData ⌃ Expand all **Object** uid: bounceData

Examples values:

{"address":"90:FD:9F:AC:9B:7F","sequenceNumber":11,"timeOfFlight":1425,"toFDelta":45,"impactTimestamp":9072,"isInvalid":false}

address **String**

Unique sensor identifier.

sequenceNumber **Integer**

The order of the bounce event relative to other events from the same sensor. This value corresponds to the sequence number of the IMPACT packet which ends the flight phase of the bounce. Note - This sequence will not be contiguous.

timeOfFlight **Integer**

Duration (ms) from DEPART to IMPACT event timestamps.

toFDelta **Integer**

Difference (ms) in timeOfFlight from the previous bounce.

impactTimestamp **Integer**

Timestamp (ms) of the IMPACT event which ends the flight phase.

isInvalid **Boolean**

Indicates that a bounceData packet which was previously broadcast is now determined to be invalid and should be discarded.

Additional properties are allowed.

idle ⌃ Expand all **Object** uid: idle

event **String**

idle

Const: "idle"

data ⌃ Expand all **Object** uid: idleData

Examples values:

{"address":"90:FD:9F:AC:9B:7F","sequenceNumber":12,"impactTimestamp":10352}

address **String**

Unique sensor identifier.

sequenceNumber **Integer**

The order of the idle event relative to other events from the same sensor.

idleTimestamp **Integer**
Timestamp (ms) of the IDLE event.

Additional properties are allowed.

Additional properties are allowed.

idleData ^ Expand all **Object** uid: idleData
Examples values:
`{"address":"90:FD:9F:AC:9B:7F","sequenceNumber":12,"impactTimestamp":10352}`

address **String**
Unique sensor identifier.

sequenceNumber **Integer**
The order of the idle event relative to other events from the same sensor.

idleTimestamp **Integer**
Timestamp (ms) of the IDLE event.

Additional properties are allowed.

sensor-reset ^ Expand all **Object** uid: sensor-reset

event **String**
sensor-reset
Const: "sensor-reset"

data ^ Expand all **Object**
Placeholder for future development
Const: null

Additional properties are allowed.

Additional properties are allowed.

missing-data ^ Expand all **Object** uid: missing-data

event **String**
The app has detected it is missing some data from the sensors.
Const: "missing-data"

data ^ Expand all **Object** uid: missingData
Examples values:
`{"address":"90:FD:9F:AC:9B:7F","missingSequenceNumbes":[8,10]}`

address **String**

Unique sensor identifier.

missingSequenceNumbes ^ Expand all **Array<Any>**

List of currently missing sequence numbers. If this is empty it means that all previously missing packets have been found.

Additional properties are allowed.

Additional properties are allowed.

missingData ^ Expand all **Object** uid: missingData

Examples values: {"address":"90:FD:9F:AC:9B:7F","missingSequenceNumbes":[8,10]}

address **String**

Unique sensor identifier.

missingSequenceNumbes ^ Expand all **Array<Any>**

List of currently missing sequence numbers. If this is empty it means that all previously missing packets have been found.

Additional properties are allowed.

sensor-request ^ Expand all **Object** uid: sensor-request

request **String**

sensor-list

Const: "sensor-list"

Additional properties are allowed.

sensor-data ^ Expand all **Object** uid: sensor-data

Examples values: {"request":"sensor-data","args":"90:FD:9F:AC:9B:7F"}

request **String**

sensor-details

Const: "sensor-details"

args **String**

The address of a known sensor. Use the sensor-list request to get a list of valid options.

Additional properties are allowed.

sensor-list ^ Expand all **Object** uid: sensor-list

event **String**
sensor-list
Const: "sensor-list"

data ^ Expand all **Object** uid: sensorListData
Examples values: {"sensors":["90:FD:9F:AC:9B:7F","CC:CC:CC:9B:D8:FD"]}

sensors ^ Expand all **Array<Any>**
List of sensors that have been seen by the app while the Portal has been active.

Additional properties are allowed.

Additional properties are allowed.

sensorListData ^ Expand all **Object** uid: sensorListData
Examples values: {"sensors":["90:FD:9F:AC:9B:7F","CC:CC:CC:9B:D8:FD"]}

sensors ^ Expand all **Array<Any>**
List of sensors that have been seen by the app while the Portal has been active.

Additional properties are allowed.

sensor-details ^ Expand all **Object** uid: sensor-details

event **String**
sensor-details
Const: "sensor-details"

data ^ Expand all **Object** uid: sensorDetailData
Examples values:
{"friendlyName":"Trampoline 1","firmwareVersion":"1.13.3","batteryLevel":1503,"isPaired":false,"syncID":-1,"bootNumber":173,"lifetimeJumps":4542,"lifetimeAge":188678,"uniqueID":844383776,"advertisingRate":13.0791388,"isHighPower":false}

friendlyName **String**
User friendly name given to the sensor and displayed throughout Veriflite

firmwareVersion **String**
Firmware version running on the sensor

batteryLevel **Integer**
Battery voltage in mV

isPaired **Boolean**

Whether the sensor is paired (and clock synced) with another sensor. Useful for synchro calculations.

syncID **Integer**
Given to a sensor for synchro pairing. Sensors will attempt to pair with others on the same sync ID.

bootNumber **Integer**
Number of times the sensor has been booted up in its lifetime.

lifetimeJumps **Integer**
Total number of bounce events sent by the sensor in its lifetime.

lifetimeAge **Integer**
Total time the sensor has been active in its lifetime, measured in seconds.

uniqueID **Integer**
A unique ID number for a given sensor.

advertisingRate **Number**
Rate at which packets from the sensor are received by the app - indicates the strength of the signal from the sensor to app.

isHighPower **Boolean**
Whether the sensor is in high power mode (sending out packets at a higher rate).

Additional properties are allowed.

Additional properties are allowed.

sensorDetailData ^ **Object** uid: sensorDetailData

Expand all

Examples values:

{"friendlyName":"Trampoline 1","firmwareVersion":"1.13.3","batteryLevel":1503,"isPaired":false,"syncID":-1,"bootNumber":173,"lifetimeJumps":4542,"lifetimeAge":188678,"uniqueID":844383776,"advertisingRate":13.0791388,"isHighPower":false}

friendlyName **String**
User friendly name given to the sensor and displayed throughout Veriflite

firmwareVersion **String**
Firmware version running on the sensor

batteryLevel **Integer**
Battery voltage in mV

isPaired **Boolean**
Whether the sensor is paired (and clock synced) with another sensor. Useful for synchro calculations.

syncID **Integer**

Given to a sensor for synchro pairing. Sensors will attempt to pair with others on the same sync ID.

bootNumber **Integer**

Number of times the sensor has been booted up in its lifetime.

lifetimeJumps **Integer**

Total number of bounce events sent by the sensor in its lifetime.

lifetimeAge **Integer**

Total time the sensor has been active in its lifetime, measured in seconds.

uniqueID **Integer**

A unique ID number for a given sensor.

advertisingRate **Number**

Rate at which packets from the sensor are received by the app - indicates the strength of the signal from the sensor to app.

isHighPower **Boolean**

Whether the sensor is in high power mode (sending out packets at a higher rate).

Additional properties are allowed.

battery ^ Expand all **Object** uid: battery

event **String**

battery

Const: "battery"

data ^ Expand all **Object** uid: batteryData

Examples values: {"address":"90:FD:9F:AC:9B:7F","batteryLevel":1503}

address **String**

Unique sensor identifier.

batteryLevel **Integer**

Battery voltage in mV

Additional properties are allowed.

Additional properties are allowed.

batteryData ^ Expand all **Object** uid: batteryData

Examples values: {"address":"90:FD:9F:AC:9B:7F","batteryLevel":1503} *address*

String

Unique sensor identifier.

batteryLevel

Integer

Battery voltage in mV

Additional properties are allowed.

status-flags ^ Expand all **Object** uid: status-flags

event

String

status-flags

Const: "status-flags"

data ^ Expand all **Object** uid: statusFlagData

Examples values:

{"address":"90:FD:9F:AC:9B:7F","uniqueID":844383776,"firmwareVersion":"1.13.3","isHighPower":false,"bootNumber":134,"lifetimeJumps":4542,"lifetimeAge":188678}

address

String

Unique sensor identifier.

uniqueID

Integer

A unique ID number for a given sensor.

firmwareVersion

String

Firmware version running on the sensor.

isHighPower

Boolean

Whether the sensor is in high power mode (sending out packets at a higher rate).

bootNumber

Integer

Number of times the sensor has been booted up in its lifetime.

lifetimeJumps

Integer

Total number of bounce events ever detected by the sensor.

lifetimeAge

Integer

Total time the sensor has been active in its lifetime, measured in seconds.

Additional properties are allowed.

Additional properties are allowed.

statusFlagData ^ Expand all **Object** uid: statusFlagData

Examples values: `{"address":"90:FD:9F:AC:9B:7F","uniqueID":844383776,"firmwareVersion":"1.13.3","isHighPower":false,"bootNumber":134,"lifetimeJumps":4542,"lifetimeAge":188678}`

address
String
Unique sensor identifier.

uniqueID
Integer
A unique ID number for a given sensor.

firmwareVersion
String
Firmware version running on the sensor.

isHighPower
Boolean
Whether the sensor is in high power mode (sending out packets at a higher rate).

bootNumber
Integer
Number of times the sensor has been booted up in its lifetime.

lifetimeJumps
Integer
Total number of bounce events ever detected by the sensor.

lifetimeAge
Integer
Total time the sensor has been active in its lifetime, measured in seconds.

Additional properties are allowed.

display-name ^ Expand all **Object** uid: display-name

event
String
display-name
Const: "display-name"

data ^ Expand all
Object uid: displayNameData
Examples values: `{"address":"90:FD:9F:AC:9B:7F","friendlyName":"Trampoline 1"}`

address
String
Unique sensor identifier.

friendlyName
String
User friendly name given to the sensor and displayed throughout Veriflite.

Additional properties are allowed.

Additional properties are allowed.

displayNameData ^ Expand all **Object** uid: displayNameData

Examples values: {"address":"90:FD:9F:AC:9B:7F","friendlyName":"Trampoline 1"}

address

String

Unique sensor identifier.

friendlyName

String

User friendly name given to the sensor and displayed throughout Veriflite.

Additional properties are allowed.