Smarter Technologies

API Developers Guide

Version 1.1

8th June 2021

CYBER ESSENTIALS

Microsoft Silver Partner

IASME Consortium®

Crown Commercial Service *Supplier*

1 Austin Park, Yeoman Road, Ringwood, Hampshire, BH24 3FG, United Kingdom

Phone: +44 (0)330 223 5000
Email: enquiries@smartertechnologies.com
Web: www.smartertechnologies.com

Table of Contents

Introduction **6**
Service Overview *6*
API Standards *6*
HTTP Methods 6
HTTP Status Codes 6
Passing Data to the API Service 7
Output Formats 7
Dates 7
HTTPS 7
Fair-Use Policy 7
Questions / Problems 7
Terminology **8**
Accessing the Service **9**
Push Service **10**
Endpoint / Webhook Requirements *10*
Sample Endpoints *10*
Formatted and Raw Event Samples *10*
Formatted Example 11
Raw Example 11
Tag Type JSON Data Specifications *11*
API Functions Overview **12**
API Service URL *12*
Sending your API key *12*
Specifying Output Format *12*
Querying Data *12*
Sorting 12
Searching 13
Filtering 13
Paging 14
Tag API **15**
Get Tag *15*
URL Endpoint 15
HTTP Method 15
Request Parameters 15

Request Format 15
Request Content 15
Expected Response Codes 15
Response Format 15
Response Fields 15
Example Response 16

Create Tag *17*
URL Endpoint 17
HTTP Method 17
Request Parameters 17
Request Format 17
Request Content 17
Example Request 17
Expected Response Codes 17
Response Format 18
Response Fields 18
Example Response 18

Delete Tag *19*
URL Endpoint 19
HTTP Method 19
Request Parameters 19
Request Format 19
Request Content 19
Expected Response Codes 19
Response Format 19
Response Fields 19

List Tag Events *20*
URL Endpoint 20
HTTP Method 20
Search Fields 20
Ordering Fields 20
Filter Fields / Options Supported 20
Raw Format Supported? 21
Response Format 21
Expected Response Codes 21
Example Response 21
Raw Response Example 25

List Events for Tag *28*
URL Endpoint 28
HTTP Method 28
Search Fields 28
Ordering Fields 28
Filter Fields / Options Supported 28
Raw Format Supported? 29
Response Format 29
Expected Response Codes 29
Example Response 29
Raw Response Example 33

List Tag Types *36*
URL Endpoint 36
HTTP Method 36
Expected Response Codes 36
Response Format 36
Response Fields 36
Example Response 36

Create Tag Command *38*
Note 38
URL Endpoint 38
HTTP Method 38
Request Parameters 38
Request Format 39
Request Content 39
Expected Response Codes 39
Response Format 39
Response Fields 39
Example Response 40

Get Tag Command *41*
Note 41
URL Endpoint 41
HTTP Method 41
Expected Response Codes 41
Response Format 41
Response Fields 41
Example Response 42

BB5 API 43

Get BB5 ..... *43*
URL Endpoint 43
HTTP Method 43
Request Parameters 43
Request Format 43
Request Content 43
Expected Response Codes 43
Response Format 43
Response Fields 43
Example Response 44

List BB5 Events *45*
URL Endpoint 45
HTTP Method 45
Search Fields 45
Ordering Fields 45
Filter Fields / Options Supported 45
Raw Format Supported? 46
Response Format 46
Expected Response Codes 46
Example Response 46

Raw Response Example 50

List Events for Tag *53*
URL Endpoint 53
HTTP Method 53
Search Fields 53
Ordering Fields 53
Filter Fields / Options Supported 53
Raw Format Supported? 53
Response Format 53
Expected Response Codes 54
Example Response 54
Raw Response Example 58

} ..... Error! Bookmark not defined.

Appendix A – Tag Types and Event Examples 59

Common Data Fields *59*

Bolus *59*
Example Event Data 59

Contact (e.g. Window / Defibrillator) *60*
Example Event Data 60

GPS *60*
Example Event Data 60

UART Tags *61*
Pulse Only 61
Example Event Data 61
Legionella 61
Example Event Data 61
Pipe Sensor 62
Example Event Data 62
Bin / Depth Sensor 62
Example Event Data 62
Air Quality 62
Example Event Data 62
Electricity Contact Pulse Meter Readers 64
Example Event Data 64
Sound 65
Example Event Data 65

Introduction

Service Overview

The Smarter Technologies Group API allows you to integrate Smarter Technologies directly into your website or internal systems. It is a REST-based API, which means that it is easy to integrate with and consume. It has been designed to provide you with the means of integrating nearly every function of Smarter Technologies into your own systems.

The API is designed around the principles of pushing and pulling data. Using a pub/sub model, data can be consumed either by setting up a Webhook (an HTTP endpoint) to which the servce will publish data and details of events as they are generated (in a pub-sub model), or by a standard RESTful interface which allows data to be queried or sent to the service (i.e. standard CRUD operations).

This document provides detailed instructions on how to integrate with the API, including:

- How to gain access to the API, including requesting API keys, and configuring endpoints as Webhooks for the Push Service.
- Instructions on how to connect to the API, including passing the relevant security credentials.
- Overview of events published by the Push Service.
- Detailed description of each API function, with examples of relevant requests and responses.

API Standards

HTTP Methods

The Smarter Technologies Group API is a RESTful API. At a basic level, it uses the following HTTP methods to perform different actions:

HTTP Method	Action
GET	Retrieves a single record, or list of records
POST	Creates a new record
PUT	Updates a single record
DELETE	Deletes a single record

HTTP Status Codes

Standard HTTP status codes are used in response to requests made to the API service. They are:

HTTP Status Code	Meaning
404	Either the single record you are trying to retrieve is missing, or you have tried to call a non-existant function or endpoint.
400	Incorrect information has been sent to the service; more information will be shown in the error itself contained in the response body.

401	You have not provided the relevant / correct security credentials in the request to the service.
403	You are not authorised to access the function.
200	The request was successful.

Passing Data to the API Service

All data passed to the service must be in JSON string format either via a PUT or POST operation depending on the function called. The documentation for each function will describe to you what parameters are required to be supplied (either directly in the URL, for example */Tags/000123456A, or as a query string parameter, for example ?skip=0&take=10*.

Output Formats

Currently, all data output by the service, either via Webhooks or via API endpoints, will be in JSON string format.

Dates

Note that all dates are stored and returned as UTC (GB +0:00).

HTTPS

All request to API endpoints must be via HTTPS. Requests made via HTTP will be ignored.

Fair-Use Policy

Access to the Smarter Technologies Group API is controlled via a fair-use policy. Requests to the service will be limited to 5 requests per second per IP address. This limit is in place to protect the API and other users from potential abuse and disruptive behaviour. Persistent contravention of the fair-use policy may result in you being blocked from accessing the service.

Questions / Problems

If you are having problems with the Smarter Technologies API, or if you have any questions, please contact the Smarter Technologies Service Desk.

Terminology

The Smarter Technologies API uses a standard set of terminology to make it easier to develop against. This terminology is detailed below:

Word	Meaning
Webhook	A URL (i.e. an HTTP endpoint) provided by a customer wishing to integrate with the Push Service, to which data and events will be published instantly as they occur.
API Endpoint	The point at which a consuming service or application can interact directly with the API to gain access to data or the underlying function. The endpoint is essentially the URL used to send requests to the service.
Device	A generic term for a component that reports data into the Smarter Technologies service. Gateways and Tags are examples of devices.
Gateway	Part of the Orion network, gateways collect and send data from devices such as tags to the Smarter Technologies service. Gateways are also used to deliver commands to Tags.
Tag	A device that essentially collects data from attached sensors and sends on to gateways when they are in range.
BB5	GPS Tracking device which can also detect motion and tilt.
Event	An event occurs when a device such as a Gateway sends data to the service. Events can be generated by Gateways, Tags, and also by the service itself.
Command	Certain devices, such as Gateways and Tags, can be sent commands to perform operations, or update configuration settings.
Alert	Alerts can be configured to be raised when certain criteria are met, for example a temperature reached. Alerts are generally broadcast via Webhooks.
Schedule	The means by which devices such as Tags can be scheduled to perform operations via commands at certain times.

Accessing the Service

To gain access to the Smarter Technologies service, please contact the Smarter Technologies Service Desk. You will be provided with an API key in order to access the API endpoints, and you are also able to provide a URL endpoint in order to subscribe to the Smarter Technologies API Push Service.

A web interface will shortly be provided whereby you will be able to login to your account and generate API keys and configure webhook URLs without the need to contact the Service Desk.

Push Service

The API service provides a Push Service, essentially a publisher / subscriber model, whereby customers are able to provide a URL to which data generated by their devices is published. By subscribing to the API service in this manner, your systems will receive the required data instantly, without having to poll the API service unnecessarily.

The Push Service broadcasts data and events by POSTing the information in JSON format to a URL you have provided to the Smarter Technologies Service Desk. This endpoint or webhook must therefore support HTTP POST requests.

The endpoint can be configured to specify information such as API key in headers of your choosing for each request it attempts to make to your endpoint. If you require an API key or any other to be sent to the endpoint, please let the Smarter Technologies Service Desk know when you are requesting access to the service.

Finally, you can request data to be sent in a formatted format (i.e. the data sent by a device is parsed and sent in a formatted structure), or in raw format (i.e. the data sent by a device is not parsed, and is sent in its raw format). Samples explaining this are provided below. Again, please let the Smarter Technologies Service Desk know which format you would prefer to receive data.

Endpoint / Webhook Requirements

Your endpoint must support HTTP POST requests, and accept JSON data.

Note that the Push Service will only attempt to send the data once. If the endpoint is not available or errors, the service will not try again.

Sample Endpoints

Smarter Technologies have created a series of sample endpoints which demonstrate how to create basic endpoints for the receiving and processing of data from the Push Service.

These are available on GitHub here.

More endpoints will be added to cover a range of languages and frameworks.

Numerous tutorials can be found online instructing how to develop solutions to provide endpoints. For example:

https://crunchify.com/create-very-simple-jersey-rest-service-and-send-json-data-from-java-client/

Formatted and Raw Event Samples

As described above, data pushed to customer endpoints can either be in formatted, or raw output. For the majority of use cases, the formatted output will suffice.

The following are examples of formatted and raw data received for the same event generated by a Pipe Sensor tag:

Formatted Example


```
{
	"tag_id": "80200133",
	"gateway_id": "5410ECF3FAC2",
	"voltage": 3.5000,
	"temperature": "+24 to +26",
	"pin_state": "Low",
	"uart": "c2,29.3,20.1",
	"pulse": false,
	"packet_sequence_number": 4,
	"rssi": "-79",
	"alert_1": false,
	"alert_2": false,
	"extended_data": "0A0063322C32392E332C32302E31",
	"has_extended_data": true,
	"temperature_readings": {
		"hot_temperature_celsius": 29.3000,
		"cold_temperature_celsius": 20.1000
	},
	"event_id": 270194629,
	"device_id": 3710,
	"event_type": "Tag",
	"event_received_datetime_utc": "2021-06-09T11:08:30.125"
}
```


Raw Example


```
{
	"tag_id": "80200133",
	"gateway_id": "5410ECF3FAC2",
	"event_id": 270194629,
	"device_id": 3710,
	"event_type": "Tag",
	"event_received_datetime_utc": "2021-06-09T11:08:30.125",
	"raw_data": "$G,5,5410ECF3FAC2,R,15202,-79,-
	513,69B2,80200133,4,B,0,0,0,1,B7,0A0063322C32392E332C32302E31*20"
}
```


Tag Type JSON Data Specifications

The JSON format specifications for each of the types of tag supported by the Push Service are detailed in Appendix A – Tag Types and Event Examples

API Functions Overview

API Service URL

The root URL of all API endpoints is the following:

https://api.smartertechnologies.com/v3

Sending your API key

All API functions require you to send the API key provided by Smarter Technologies Service Desk in order to successfully make requests.

The API key must be passed as a header in the request, named **key.**

If this key header is not found, or its value is null or empty, the request will not be successful.

Specifying Output Format

For some of the API functions, data can be returned in one of two ways, as a structured format (default) or in a raw format. For example, when requesting to view Tag events, these can be returned in a structured format, with extended data and/or UART data parsed into specific data items, or it can be returned raw, as sent by the Tag and the Gateways themselves. In raw format, you would need to parse the data in your own system.

To receive the data in raw format, you need to add a query string value of *?format=raw* as part of the URL you construct to call the relevant endpoint.

If you do not provide a *format* query string pair, or its value is anything other than *raw*, the data will be output in a structured format.

Querying Data

Endpoints which return multiple records (typically GET requests where no ID has been supplied) can be instructed to query records in different ways, as well as return more or less records per request as required. The API service uses query string values to control this functionality. For example the following URL will return the first 100 events for Tag 8123456 in descending order of the date of the event:

/tags/8123456/events?take=100&ordering=event_received_datetime_utc-

Sorting

Sorting is achieved by providing the name of the field or fields on which you wish to sort, and the whether you wish the sort on that field to be ascending or descending in value. Only certain fields can be sorted on – the documentation for each endpoint will detail the fields that can be search on.

To specify that you wish to sort on a field in ascending order add the query string ordering followed by the name of the field you wish to sort on, for example *?ordering=event_received_datetime_utc*

To specify to sort descending, add a minus character after the name of the field, for example *?ordering=event_received_datetime_utc-*

To sort by multiple fields, supply the names of the fields (and whether in ascending or descending order) separated by commas, for example to sort by event date descending and device id ascending, it would be ?ordering=event_received_datetime_utc-,device_id

Searching

Certain endpoints allow you to supply a value to be searched on in pre-configured fields. This is done by adding a query string called *search* and supply the term you wish to search on as its value.

For example, ?search=0001234

The documentation for each endpoint will instruct you on whether there are any fields that will be searched on if this query string value is supplied.

Filtering

Certain endpoints allow you to filter the records that are retrieved on pre-configured fields. To filter by a particular field, you simply add the name of the field to the query string, with the value you wish to filter on. So, for example to filter by Device Id, you would add *?device_id=1*

Multiple filters can be applied per request.

How the filters are applied by can further modified by adding a filter option to the end of the field name, separated by a double underscore __

For example, to filter by all records where the event date is earlier than start of August 2020, you would add ?*event_received_datetime_utc__lt=2020-08-01*

The following filter options are supported by the API:

Filter Option	Description
gte	Greater Than or Equal To
gt	Greater Than
lte	Less Than or Equal To
lt	Less Than
eq	Equals (default if no filter option is specified)
neq	Not Equals
in	Equals one of (values supplied separated by commas. Note that a maximum of 10 values can be supplied.)
en	Is Null
nen	Is Not Null

Again, the documentation for each endpoint will instruct you on what fields can be filtered, and using which filter commands.

Paging

Paging of data (i.e. how many and at what starting point in a query records are returned) is controlled by two query string values, *skip* and *take*. This is particularly useful when displaying multiple records in a table. Take instructs the service to return the specified number of records, *up to a maximum of 1000*. Skip instructs the service to ignore the first *n* records it retrieves once any sorting, filtering and searching criteria have been applied.

So, for example, to return the second 100 records in a query, add ?skip=100&take=100

Tag API

Get Tag

Retrieves the details of a Tag, including latest event if data has been received

URL Endpoint

/tags/[Tag Id]

e.g. /tags/82123456

HTTP Method

GET

Request Parameters

Parameter	Description
Tag Id	The unique id of the Tag, e.g. 82123456

Request Format

application/json

Request Content

None

Expected Response Codes

HTTP Status Code	Meaning
401	You have not provided the relevant / correct security credentials in the request to the service.
403	You are not authorised to access the function.
200	The request was successful.
404	No Tag was found matching the Tag Id, or you are not entitled to view the data.

Response Format

application/json

Response Fields

Field	Child Field	Type	Description
device_id		Long	The unique identifier of the device
type		Map	The type of Tag
	id	Integer	The unique id of the tag type
	description	String	The description of the tag type
tag_id		String	The unique identifier of the Tag, e.g. 8123456
description		String	A description of the Tag
last_event		Map	The recorded event received from the Tag

Example Response

The last_event field will contain additional / different fields depending on the type of tag generating the data. Examples of the different types of Tags can be found in Appendix A. The following is an example response from a Pipe Sensor Tag.


```
{
  "device_id": 216,
  "tag_id": "80200425",
  "description": "This is a Pipe Sensor",
  "last_event": {
    "tag_id": "80200425",
    "gateway_id": "001EC0828146",
    "voltage": 3.5000,
    "temperature": "+24 to +26",
    "pin_state": "Low",
    "uart": "c2,13.8,13.5",
    "pulse": false,
    "packet_sequence_number": 1,
    "rssi": "-71",
    "extended_data": "0A0063322C31332E382C31332E35",
    "has_extended_data": true,
    "temperature_readings": {
      "hot_temperature_celsius": 13.8000,
      "cold_temperature_celsius": 13.5000
    },
    "event_id": 2126633,
    "device_id": 216,
    "event_type": "Tag",
    "event_received_datetime_utc": "2020-09-28T08:58:22.849"
  }
}
```

Create Tag

Registers a Tag with the API service

URL Endpoint

/tags

HTTP Method

POST

Request Parameters

None

Request Format

application/json

Request Content

Field	Type	Max Length	Required	Description
tag_id	String	25	Yes	The unique identifier of the Tag, e.g. 8123456
description	String	250	No	A description of the Tag
type_id	Integer		Yes	The type of tag. For list of valid tag type ids, please see List Tag Type endpoint description.

Example Request

```
{
    "tag_id": "61234567",
    "description": "This is a test tag",
    "type_id": 2
}
```

Expected Response Codes

HTTP Status Code	Meaning
401	You have not provided the relevant / correct security credentials in the request to the service.
403	You are not authorised to access the function.
200	The request was successful.
400	Invalid data was supplied: • No data supplied • tag_id field missing or empty • tag_id already registered • tag_id invalid – Max length exceeded.

Response Format

application/json

Response Fields

Field	Child Field	Type	Description
device_id		Long	The unique id of the newly created Device record
type		Map	The type of Tag
	id	Integer	The unique id of the tag type
	description	String	The description of the tag type
tag_id		String	The unique identifier of the Tag, e.g. 8123456
description		String	A description of the Tag

Example Response

```
{
  "device_id": 525,
  "type": {
    "id": 2,
    "description": "Beacon"
  },
  "tag_id": "61234567",
  "description": "This is a test tag"
}
```

Delete Tag

De-registers a Tag with the API service. Once invoked, data for the Tag will no longer be available to the customer, and will no longer be published to the customer's webhooks via the Push service.

URL Endpoint

/tags/[Tag Id]

e.g. /tags/82123456

HTTP Method

DELETE

Request Parameters

None

Request Format

application/json

Request Content

None

Expected Response Codes

HTTP Status Code	Meaning
401	You have not provided the relevant / correct security credentials in the request to the service.
403	You are not authorised to access the function.
200	The request was successful.
404	The specified tag id was not found

Response Format

application/json

Response Fields

None

List Tag Events

Retrieves a list of events generated by Tags.

URL Endpoint

/tags/events

HTTP Method

GET

Search Fields

Field	Description
tag_id	Tag Id field contains the specified value
gateway_id	Gateway Id field contains the specified value

Ordering Fields

Field	Description
event_received_datetime_utc	Order by event date
event_id	Order by event unique identifier
tag_id	Order by the tag id
gateway_id	Order by the gateway id

Filter Fields / Options Supported

Field	Data Type	Description
event_id__gt	Long	Event id is greater than the supplied value
event_id__gte	Long	Event id is greater than or equal to the supplied value
event_id__lt	Long	Event id is less than the supplied value
event_id__lte	Long	Event id is less than or equal to the supplied value
event_received_datetime_utc__gt	Time and/or Date	Event date is greater than the supplied value
event_received_datetime_utc__gte	Time and/or Date	Event date is greater than or equal to the supplied value
event_received_datetime_utc__lt	Time and/or Date	Event date is less than the supplied value
event_received_datetime_utc__lte	Time and/or Date	Event date is less than or equal to the supplied value
gateway_id__eq	String	The name of the gateway which received the data from the Tag

gateway_id__in	String values (comma-separated)	Gateway Id matches one of the supplied values. Values must be comma-separated,up to a maximum of 10 values.
tag_id__eq	String	The tag id of the event matches the supplied value. Note that __eq is not required.
tag_id__in	String values (comma-separated)	Tag ID matches one of the supplied values. Values must be comma-separated, up to a maximum of 10 values.

Raw Format Supported?

Yes

Response Format

application/json

Expected Response Codes

HTTP Status Code	Meaning
401	You have not provided the relevant / correct security credentials in the request to the service.
403	You are not authorised to access the function.
200	The request was successful.

Example Response

Each record in the "results" field will contain additional / different fields depending on the type of tag generating the data. Examples of the different types of Tags can be found in Appendix A.


```
{
    "total_record_count": 197,
    "filtered_record_count": null,
    "options": {
        "ordering": null,
        "filtering": [],
        "search": null,
        "skip": 0,
        "take": 10,
        "format": "Formatted"
    },
    "results": [
        {
            "tag_id": "81200076",
            "gateway_id": "001EC082B4FA",
            "voltage": 3.3000,
```

```
        "temperature": "+24 to +26",
        "uart": "X 814 Y -1908 Z 1499",
        "packet_sequence_number": 36,
        "rssi": "-65",
        "extended_data": "0A0058203831342059202D31393038205A2031343939",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3474302,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-11T20:15:54.992",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+21 to +23",
        "uart": "X 802 Y -1878 Z 1482",
        "packet_sequence_number": 41,
        "rssi": "-62",
        "extended_data": "0A0058203830322059202D31383738205A2031343832",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3473945,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-11T14:10:16.776",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+18 to +20",
        "uart": "X 760 Y -1839 Z 1487",
        "packet_sequence_number": 8,
        "rssi": "-62",
        "extended_data": "0A0058203736302059202D31383339205A2031343837",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3471104,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-09T13:37:55.41",
        "raw_data": null
    },
    {
```

```
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+18 to +20",
        "uart": "X 747 Y -1812 Z 1487",
        "packet_sequence_number": 49,
        "rssi": "-59",
        "extended_data": "0A0058203734372059202D31383132205A2031343837",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3464412,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-04T19:13:47.346",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+15 to +17",
        "uart": "X 738 Y -1818 Z 1479",
        "packet_sequence_number": 19,
        "rssi": "-65",
        "extended_data": "0A0058203733382059202D31383138205A2031343739",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3464175,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-04T15:11:11.792",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+15 to +17",
        "uart": "X 743 Y -1826 Z 1475",
        "packet_sequence_number": 23,
        "rssi": "-61",
        "extended_data": "0A0058203734332059202D31383236205A2031343735",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3464001,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-04T12:12:52.145",
```

```
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+18 to +20",
        "uart": "X 761 Y -1842 Z 1483",
        "packet_sequence_number": 1,
        "rssi": "-73",
        "extended_data": "0A0058203736312059202D31383432205A2031343833",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3461641,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-02T19:52:59.776",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+21 to +23",
        "uart": "X 969 Y -1863 Z 1340",
        "packet_sequence_number": 5,
        "rssi": "-53",
        "extended_data": "0A0058203936392059202D31383633205A2031333430",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3440082,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-07-29T16:23:26.803",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+18 to +20",
        "uart": "X 949 Y -1857 Z 1341",
        "packet_sequence_number": 1,
        "rssi": "-58",
        "extended_data": "0A0058203934392059202D31383537205A2031333431",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3434811,
```

```
            "device_id": 5534,
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-07-25T20:25:08.763",
            "raw_data": null
        },
        {
            "tag_id": "81200076",
            "gateway_id": "001EC082B4FA",
            "voltage": 3.3000,
            "temperature": "+21 to +23",
            "uart": "X 957 Y -1860 Z 1343",
            "packet_sequence_number": 25,
            "rssi": "-57",
            "extended_data": "0A0058203935372059202D31383630205A2031333433",
            "gps": null,
            "air_quality": null,
            "temperature_log": null,
            "bin": null,
            "event_id": 3434492,
            "device_id": 5534,
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-07-25T14:58:57.847",
            "raw_data": null
        }
    ]
}
```


Raw Response Example


```
{
    "total_record_count": 993379,
    "options": {
        "filtering": [],
        "skip": 0,
        "take": 10,
        "format": "Raw"
    },
    "results": [
        {
            "tag_id": "81100547",
            "gateway_id": "001EC0828146",
            "event_id": 2226072,
            "device_id": 94,
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-09-28T12:37:37.993",
            "raw_data": "$G,5,001EC0828146,R,18030,-58,-12,69B2,81100547,13,B,1,0,0,1,98,*22"
        },
        {
            "tag_id": "81200200",
            "gateway_id": "001EC0828146",
            "event_id": 2226071,
            "device_id": 330,
            "event_type": "Tag",
```

```
        "event_received_datetime_utc": "2020-09-28T12:37:37.993",
        "raw_data": "$G,5,001EC0828146,R,17714,-51,-12,69B2,81200200,9,B,1,0,0,1,9F,0A00*16"
    },
    {
        "tag_id": "00022759",
        "gateway_id": "001EC0828146",
        "event_id": 2226070,
        "device_id": 240,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-09-28T12:37:37.993",
        "raw_data": "$G,5,001EC0828146,R,17499,-37,297,69B2,00022759,30,B,0,0,0,1,88,*35"
    },
    {
        "tag_id": "81100589",
        "gateway_id": "001EC0828146",
        "event_id": 2226068,
        "device_id": 219,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-09-28T12:37:37.993",
        "raw_data": "$G,5,001EC0828146,R,17252,-51,-334,69B2,81100589,1,B,1,0,0,1,98,*24"
    },
    {
        "tag_id": "0002274A",
        "gateway_id": "001EC0828146",
        "event_id": 2226062,
        "device_id": 391,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-09-28T12:37:35.987",
        "raw_data": "$G,5,001EC0828146,R,16019,-60,-111,69B2,0002274A,18,B,0,0,0,1,78,*66"
    },
    {
        "tag_id": "000209DE",
        "gateway_id": "001EC0828146",
        "event_id": 2226060,
        "device_id": 149,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-09-28T12:37:35.987",
        "raw_data": "$G,5,001EC0828146,R,15300,-60,173,69B2,000209DE,6,B,1,0,0,1,A9,*76"
    },
    {
        "tag_id": "81600065",
        "gateway_id": "001EC0828146",
        "event_id": 2226058,
        "device_id": 77,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-09-28T12:37:35.987",
        "raw_data": "$G,5,001EC0828146,R,14959,-55,-37,69B2,81600065,18,B,0,0,0,1,88,0603*2E"
    },
```

```
        {
            "tag_id": "81600320",
            "gateway_id": "001EC0828146",
            "event_id": 2226057,
            "device_id": 136,
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-09-28T12:37:35.987",
            "raw_data": "$G,5,001EC0828146,R,14781,-35,-49,69B2,81600320,2,B,0,0,3,1,98,*14"
        },
        {
            "tag_id": "81600310",
            "gateway_id": "001EC0828146",
            "event_id": 2226050,
            "device_id": 179,
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-09-28T12:37:33.947",
            "raw_data": "$G,5,001EC0828146,R,13615,-31,-111,69B2,81600310,1,B,1,0,3,1,88,*27"
        },
        {
            "tag_id": "81100770",
            "gateway_id": "001EC0828146",
            "event_id": 2226049,
            "device_id": 532,
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-09-28T12:37:33.947",
            "raw_data": "$G,5,001EC0828146,R,13404,-35,12,69B2,81100770,0,B,0,0,1,1,84,*35"
        }
    ]
}
```

List Events for Tag

Retrieves a list of events generated by a particular Tag.

URL Endpoint

/tags/[Tag Id]/events

e.g. /tags/81200076/events

HTTP Method

GET

Search Fields

Field	Description
gateway_id	Gateway Id field contains the specified value

Ordering Fields

Field	Description
event_received_datetime_utc	Order by event date
event_id	Order by event unique identifier
gateway_id	Order by the gateway id

Filter Fields / Options Supported

Field	Data Type	Description
event_id__gt	Long	Event id is greater than the supplied value
event_id__gte	Long	Event id is greater than or equal to the supplied value
event_id__lt	Long	Event id is less than the supplied value
event_id__lte	Long	Event id is less than or equal to the supplied value
event_received_datetime_utc__gt	Time and/or Date	Event date is greater than the supplied value
event_received_datetime_utc__gte	Time and/or Date	Event date is greater than or equal to the supplied value
event_received_datetime_utc__lt	Time and/or Date	Event date is less than the supplied value
event_received_datetime_utc__lte	Time and/or Date	Event date is less than or equal to the supplied value
gateway_id__eq	String	The name of the gateway which received the data from the Tag

gateway_id__in	String values (comma-separated)	Gateway Id matches one of the supplied values. Values must be comma-separated,up to a maximum of 10 values.

Raw Format Supported?

Yes

Response Format

application/json

Expected Response Codes

HTTP Status Code	Meaning
401	You have not provided the relevant / correct security credentials in the request to the service.
403	You are not authorised to access the function.
200	The request was successful.
404	A tag matching the supplied ID was not found, or you are not entitled to view the record for that Tag.

Example Response

Each record in the "results" field will contain additional / different fields depending on the type of tag generating the data. Examples of the different types of Tags can be found in Appendix A.


```
{
    "total_record_count": 197,
    "filtered_record_count": null,
    "options": {
        "ordering": null,
        "filtering": [],
        "search": null,
        "skip": 0,
        "take": 10,
        "format": "Formatted"
    },
    "results": [
        {
            "tag_id": "81200076",
            "gateway_id": "001EC082B4FA",
            "voltage": 3.3000,
            "temperature": "+24 to +26",
            "uart": "X 814 Y -1908 Z 1499",
            "packet_sequence_number": 36,
            "rssi": "-65",
            "extended_data": "0A005820383134205920 2D31393038205A2031343939",
            "gps": null,
            "air_quality": null,
```

```
        "temperature_log": null,
        "bin": null,
        "event_id": 3474302,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-11T20:15:54.992",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+21 to +23",
        "uart": "X 802 Y -1878 Z 1482",
        "packet_sequence_number": 41,
        "rssi": "-62",
        "extended_data": "0A0058203830322059202D31383738205A2031343832",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3473945,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-11T14:10:16.776",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+18 to +20",
        "uart": "X 760 Y -1839 Z 1487",
        "packet_sequence_number": 8,
        "rssi": "-62",
        "extended_data": "0A0058203736302059202D31383339205A2031343837",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3471104,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-09T13:37:55.41",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+18 to +20",
        "uart": "X 747 Y -1812 Z 1487",
        "packet_sequence_number": 49,
        "rssi": "-59",
```

```
        "extended_data": "0A0058203734372059202D31383132205A2031343837",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3464412,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-04T19:13:47.346",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+15 to +17",
        "uart": "X 738 Y -1818 Z 1479",
        "packet_sequence_number": 19,
        "rssi": "-65",
        "extended_data": "0A0058203733382059202D31383138205A2031343739",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3464175,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-04T15:11:11.792",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+15 to +17",
        "uart": "X 743 Y -1826 Z 1475",
        "packet_sequence_number": 23,
        "rssi": "-61",
        "extended_data": "0A0058203734332059202D31383236205A2031343735",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3464001,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-04T12:12:52.145",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+18 to +20",
```

```
        "uart": "X 761 Y -1842 Z 1483",
        "packet_sequence_number": 1,
        "rssi": "-73",
        "extended_data": "0A0058203736312059202D31383432205A2031343833",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3461641,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-02T19:52:59.776",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+21 to +23",
        "uart": "X 969 Y -1863 Z 1340",
        "packet_sequence_number": 5,
        "rssi": "-53",
        "extended_data": "0A0058203936392059202D31383633205A2031333430",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3440082,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-07-29T16:23:26.803",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+18 to +20",
        "uart": "X 949 Y -1857 Z 1341",
        "packet_sequence_number": 1,
        "rssi": "-58",
        "extended_data": "0A0058203934392059202D31383537205A2031333431",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3434811,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-07-25T20:25:08.763",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
```

```
            "gateway_id": "001EC082B4FA",
            "voltage": 3.3000,
            "temperature": "+21 to +23",
            "uart": "X 957 Y -1860 Z 1343",
            "packet_sequence_number": 25,
            "rssi": "-57",
            "extended_data": "0A0058203935372059202D31383630205A2031333433",
            "gps": null,
            "air_quality": null,
            "temperature_log": null,
            "bin": null,
            "event_id": 3434492,
            "device_id": 5534,
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-07-25T14:58:57.847",
            "raw_data": null
        }
    ]
}
```


Raw Response Example


```
{
    "total_record_count": 1942,
    "options": {
        "filtering": [],
        "skip": 0,
        "take": 10,
        "format": "Raw"
    },
    "results": [
        {
            "tag_id": "81100547",
            "gateway_id": "001EC0828146",
            "event_id": 2226802,
            "device_id": 94,
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-09-28T12:39:40.736",
            "raw_data": "$G,5,001EC0828146,R,7438,-57,-24,69B2,81100547,14,B,0,0,0,1,88,*1D"
        },
        {
            "tag_id": "81100547",
            "gateway_id": "001EC0828146",
            "event_id": 2226072,
            "device_id": 94,
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-09-28T12:37:37.993",
            "raw_data": "$G,5,001EC0828146,R,18030,-58,-12,69B2,81100547,13,B,1,0,0,1,98,*22"
        },
        {
            "tag_id": "81100547",
            "gateway_id": "001EC0828146",
```

```
        "event_id": 2225345,
        "device_id": 94,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-09-28T12:35:37.131",
        "raw_data": "$G,5,001EC0828146,R,28423,-56,-12,69B2,81100547,12,B,0,0,0,1,98,*29"
    },
    {
        "tag_id": "81100547",
        "gateway_id": "001EC0828146",
        "event_id": 2223179,
        "device_id": 94,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-09-28T12:29:35.735",
        "raw_data": "$G,5,001EC0828146,R,62187,-55,-24,69B2,81100547,9,B,1,0,0,1,88,*10"
    },
    {
        "tag_id": "81100547",
        "gateway_id": "001EC0828146",
        "event_id": 2222438,
        "device_id": 94,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-09-28T12:27:34.857",
        "raw_data": "$G,5,001EC0828146,R,8532,-52,-37,69B2,81100547,8,B,0,0,0,1,88,*23"
    },
    {
        "tag_id": "81100547",
        "gateway_id": "001EC0828146",
        "event_id": 2221710,
        "device_id": 94,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-09-28T12:25:34.068",
        "raw_data": "$G,5,001EC0828146,R,19720,-54,0,69B2,81100547,7,B,1,0,0,1,98,*02"
    },
    {
        "tag_id": "81100547",
        "gateway_id": "001EC0828146",
        "event_id": 2220987,
        "device_id": 94,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-09-28T12:23:34.699",
        "raw_data": "$G,5,001EC0828146,R,31105,-54,-37,69B2,81100547,6,B,0,0,0,1,88,*11"
    },
    {
        "tag_id": "81100547",
        "gateway_id": "001EC0828146",
        "event_id": 2219556,
        "device_id": 94,
        "event_type": "Tag",
```

```
            "event_received_datetime_utc": "2020-09-28T12:19:36.071",
            "raw_data": "$G,5,001EC0828146,R,54277,-59,161,69B2,81100547,4,B,0,0,0,1,88,*04"
        },
        {
            "tag_id": "81100547",
            "gateway_id": "001EC0828146",
            "event_id": 2218792,
            "device_id": 94,
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-09-28T12:17:32.893",
            "raw_data": "$G,5,001EC0828146,R,65363,-59,-49,69B2,81100547,3,B,1,0,0,1,98,*13"
        },
        {
            "tag_id": "81100547",
            "gateway_id": "001EC0828146",
            "event_id": 2218074,
            "device_id": 94,
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-09-28T12:15:31.967",
            "raw_data": "$G,5,001EC0828146,R,10321,-55,0,69B2,81100547,2,B,0,0,0,1,98,*0B"
        }
    ]
}
```

List Tag Types

Retrieves the list of currently supported Tag Types.

URL Endpoint

/tags/types

HTTP Method

GET

Expected Response Codes

HTTP Status Code	Meaning
401	You have not provided the relevant / correct security credentials in the request to the service.
403	You are not authorised to access the function.
200	The request was successful.

Response Format

application/json

Response Fields

The method returns an array of the following type:

Field	Type	Description
id	Integer	The unique id of the Tag Type
description	String	A description of the tag type

Example Response

```
[
    {
        "id": 8,
        "description": "Air Quality"
    },
    {
        "id": 2,
        "description": "Beacon"
    },
    {
        "id": 6,
        "description": "Bolus"
    },
    {
        "id": 5,
        "description": "Contact"
    },
```

```
    {
        "id": 9,
        "description": "Depth"
    },
    {
        "id": 7,
        "description": "Gateway"
    },
    {
        "id": 4,
        "description": "GPS"
    },
    {
        "id": 11,
        "description": "Legionella"
    },
    {
        "id": 13,
        "description": "Pipe Sensor"
    },
    {
        "id": 10,
        "description": "Sound"
    },
    {
        "id": 1,
        "description": "Standard"
    },
    {
        "id": 12,
        "description": "Temperature"
    },
    {
        "id": 3,
        "description": "UART"
    }
]
```

Create Tag Command

Creates a command to be sent to a Tag device via an Orion Gateway.

Note

Commands are sent to Tags via the Orion Network Gateways. When a Gateway reports in, a check is made to see if there are any gateway and/or tag commands to be sent to that particular gateway. Tag commands will be sent to that Gateway if the following criteria are true:

- The Gateway has been in contact with the Tag within the last 30 days
- The Gateway has not been sent that command record before (as in that particular record matched by id)
- The Tag Command record was created within the last hour (the **created_date_utc** field)

A child record is created every time a Tag Command is sent to a particular Gateway – these are returned in the **gateways_sent** array.

URL Endpoint

/tags/[Tag Id]/commands/[Command Name]

e.g. /tags/61234567/commands/flush

HTTP Method

POST

Request Parameters

Parameter	Description
Tag Id	The unique id of the Tag, e.g. 82123456
Command Name	The name of the command to be sent to the Tag Possible values are: flush uarthigh uartlow beaconOn60m beaconOn30m beaconOn10h beaconOn20h beaconOn1w off gpsOn20m gpsOn60m gpsOn10h gpsOn20h gpsOn1w allOn

	cmd1 cmd2 cmd3 cmd4 cmd5 cmd6 cmd7 cmd8

Request Format

application/json

Request Content

None Required

Expected Response Codes

HTTP Status Code	Meaning
401	You have not provided the relevant / correct security credentials in the request to the service.
403	You are not authorised to access the function.
200	The request was successful.
404	A tag matching the supplied ID was not found, or you are not entitled to create a command record for that Tag. This status code will also be returned if a record matching the Command Name is not found

Response Format

application/json

Response Fields

Field	Sub Field	Type	Description
id		Long	The unique id of the Tag Command record
created_date_utc		DateTime	The date and time the Tag Command was created in UTC
tag_id		String	The unique identifier of the Tag
device_id		Long	The unique id of the device (i.e. the Tag)
gateways_sent_count		Integer	The number of gateways which have been sent the command
gateways_sent		Array	Contains details of any gateways which have been sent the command (i.e. if gateways_sent_count > 0)

	device_id	Long	The unique id of the device (i.e. the Gateway)
	gateway_id	String	The unique identifier of the Gateway
	sent_date_utc	DateTime	The date and time the command was sent to this gateway

Example Response


```
{
    "id": 38,
    "created_date_utc": "2020-09-22T09:55:02.9060937Z",
    "tag_id": "61234567",
    "device_id": 748,
    "gateways_sent_count": 0,
    "gateways_sent": null
}
```

Get Tag Command

Retrieves the record of an individual command created for Tag.

Note

Commands are sent to Tags via the Orion Network Gateways. When a Gateway reports in, a check is made to see if there are any gateway and/or tag commands to be sent to that particular gateway. Tag commands will be sent to that Gateway if the following criteria are true:

- The Gateway has been in contact with the Tag within the last 30 days
- The Gateway has not been sent that command record before (as in that particular record matched by id)
- The Tag Command record was created within the last hour (the **created_date_utc** field)

A child record is created every time a Tag Command is sent to a particular Gateway – these are returned in the **gateways_sent** array.

URL Endpoint

/tags/commands/[Command Id]

e.g. /tags/commands/14

HTTP Method

GET

Expected Response Codes

HTTP Status Code	Meaning
401	You have not provided the relevant / correct security credentials in the request to the service.
403	You are not authorised to access the function.
200	The request was successful.
404	A record matching the specified id parameter was not found, or you are not entitled to view that record.

Response Format

application/json

Response Fields

Field	Sub Field	Type	Description
id		Long	The unique id of the Tag Command record
created_date_utc		DateTime	The date and time the Tag Command was created in UTC

tag_id		String	The unique identifier of the Tag
device_id		Long	The unique id of the device (i.e. the Tag)
gateways_sent_count		Integer	The number of gateways which have been sent the command
gateways_sent		Array	Contains details of any gateways which have been sent the command (i.e. if gateways_sent_count > 0)
	device_id	Long	The unique id of the device (i.e. the Gateway)
	gateway_id	String	The unique identifier of the Gateway
	sent_date_utc	DateTime	The date and time the command was sent to this gateway

Example Response


```
{
    "id": 14,
    "created_date_utc": "2020-09-03T21:11:42.0309308Z",
    "tag_id": "81200093",
    "device_id": 6285,
    "gateways_sent_count": 1,
    "gateways_sent": [
        {
            "device_id": 3,
            "gateway_id": "001EC08266B3",
            "sent_date_utc": "2020-09-03T21:21:28.0033333"
        }
    ]
}
```

BB5 API

Get BB5

Retrieves the details of a BB5, including latest event if data has been received

URL Endpoint

/bb5s/[IMEI]

e.g. /bb5s/355001090607598

HTTP Method

GET

Request Parameters

Parameter	Description
IMEI	The unique id of the BB5, e.g. 355001090607598

Request Format

application/json

Request Content

None

Expected Response Codes

HTTP Status Code	Meaning
401	You have not provided the relevant / correct security credentials in the request to the service.
403	You are not authorised to access the function.
200	The request was successful.
404	No Tag was found matching the Tag Id, or you are not entitled to view the data.

Response Format

application/json

Response Fields

Field	Child Field	Type	Description
device_id		Long	The unique identifier of the device
type		Map	The type of Tag
	id	Integer	The unique id of the tag type
	description	String	The description of the tag type
tag_id		String	The unique identifier of the Tag, e.g. 8123456
description		String	A description of the Tag
last_event		Map	The recorded event received from the Tag

Example Response

The last_event field will contain additional / different fields depending on the type of tag generating the data. Examples of the different types of Tags can be found in Appendix A. The following is an example response from a Pipe Sensor Tag.


```
{
  "device_id": 216,
  "tag_id": "80200425",
  "description": "This is a Pipe Sensor",
  "last_event": {
    "tag_id": "80200425",
    "gateway_id": "001EC0828146",
    "voltage": 3.5000,
    "temperature": "+24 to +26",
    "pin_state": "Low",
    "uart": "c2,13.8,13.5",
    "pulse": false,
    "packet_sequence_number": 1,
    "rssi": "-71",
    "extended_data": "0A0063322C31332E382C31332E35",
    "has_extended_data": true,
    "temperature_readings": {
      "hot_temperature_celsius": 13.8000,
      "cold_temperature_celsius": 13.5000
    },
    "event_id": 2126633,
    "device_id": 216,
    "event_type": "Tag",
    "event_received_datetime_utc": "2020-09-28T08:58:22.849"
  }
}
```

List BB5 Events

Retrieves a list of events generated by BB5s.

URL Endpoint

/bb5s/events

HTTP Method

GET

Search Fields

Field	Description
imei	Tag Id field contains the specified value

Ordering Fields

Field	Description
event_received_datetime_utc	Order by event date
event_id	Order by event unique identifier
imei	Order by the tag id

Filter Fields / Options Supported

Field	Data Type	Description
event_id__gt	Long	Event id is greater than the supplied value
event_id__gte	Long	Event id is greater than or equal to the supplied value
event_id__lt	Long	Event id is less than the supplied value
event_id__lte	Long	Event id is less than or equal to the supplied value
event_received_datetime_utc__gt	Time and/or Date	Event date is greater than the supplied value
event_received_datetime_utc__gte	Time and/or Date	Event date is greater than or equal to the supplied value
event_received_datetime_utc__lt	Time and/or Date	Event date is less than the supplied value
event_received_datetime_utc__lte	Time and/or Date	Event date is less than or equal to the supplied value
imei__eq	String	The IMEI of the event matches the supplied value. Note that __eq is not required.

imei__in	String values (comma-separated)	IMEI matches one of the supplied values. Values must be comma-separated, up to a maximum of 10 values.

Raw Format Supported?

Yes

Response Format

application/json

Expected Response Codes

HTTP Status Code	Meaning
401	You have not provided the relevant / correct security credentials in the request to the service.
403	You are not authorised to access the function.
200	The request was successful.

Example Response


```
{
    "total_record_count": 197,
    "filtered_record_count": null,
    "options": {
        "ordering": null,
        "filtering": [],
        "search": null,
        "skip": 0,
        "take": 10,
        "format": "Formatted"
    },
    "results": [
        {
            "tag_id": "81200076",
            "gateway_id": "001EC082B4FA",
            "voltage": 3.3000,
            "temperature": "+24 to +26",
            "uart": "X 814 Y -1908 Z 1499",
            "packet_sequence_number": 36,
            "rssi": "-65",
            "extended_data": "0A005820383134205920 2D31393038205A2031343939",
            "gps": null,
            "air_quality": null,
            "temperature_log": null,
            "bin": null,
            "event_id": 3474302,
            "device_id": 5534,
            "event_type": "Tag",
```

```
        "event_received_datetime_utc": "2020-08-11T20:15:54.992",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+21 to +23",
        "uart": "X 802 Y -1878 Z 1482",
        "packet_sequence_number": 41,
        "rssi": "-62",
        "extended_data": "0A0058203830322059202D31383738205A2031343832",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3473945,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-11T14:10:16.776",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+18 to +20",
        "uart": "X 760 Y -1839 Z 1487",
        "packet_sequence_number": 8,
        "rssi": "-62",
        "extended_data": "0A0058203736302059202D31383339205A2031343837",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3471104,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-09T13:37:55.41",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+18 to +20",
        "uart": "X 747 Y -1812 Z 1487",
        "packet_sequence_number": 49,
        "rssi": "-59",
        "extended_data": "0A0058203734372059202D31383132205A2031343837",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
```

```
        "event_id": 3464412,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-04T19:13:47.346",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+15 to +17",
        "uart": "X 738 Y -1818 Z 1479",
        "packet_sequence_number": 19,
        "rssi": "-65",
        "extended_data": "0A0058203733382059202D31383138205A2031343739",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3464175,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-04T15:11:11.792",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+15 to +17",
        "uart": "X 743 Y -1826 Z 1475",
        "packet_sequence_number": 23,
        "rssi": "-61",
        "extended_data": "0A0058203734332059202D31383236205A2031343735",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3464001,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-04T12:12:52.145",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+18 to +20",
        "uart": "X 761 Y -1842 Z 1483",
        "packet_sequence_number": 1,
        "rssi": "-73",
        "extended_data": "0A0058203736312059202D31383432205A2031343833",
        "gps": null,
```

```
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3461641,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-02T19:52:59.776",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+21 to +23",
        "uart": "X 969 Y -1863 Z 1340",
        "packet_sequence_number": 5,
        "rssi": "-53",
        "extended_data": "0A0058203936392059202D31383633205A2031333430",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3440082,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-07-29T16:23:26.803",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+18 to +20",
        "uart": "X 949 Y -1857 Z 1341",
        "packet_sequence_number": 1,
        "rssi": "-58",
        "extended_data": "0A0058203934392059202D31383537205A2031333431",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3434811,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-07-25T20:25:08.763",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+21 to +23",
        "uart": "X 957 Y -1860 Z 1343",
        "packet_sequence_number": 25,
```

```
            "rssi": "-57",
            "extended_data": "0A005820393537205920 2D31383630205A2031333433",
            "gps": null,
            "air_quality": null,
            "temperature_log": null,
            "bin": null,
            "event_id": 3434492,
            "device_id": 5534,
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-07-25T14:58:57.847",
            "raw_data": null
        }
    ]
}
```


Raw Response Example


```
{
    "total_record_count": 993379,
    "options": {
        "filtering": [],
        "skip": 0,
        "take": 10,
        "format": "Raw"
    },
    "results": [
        {
            "tag_id": "81100547",
            "gateway_id": "001EC0828146",
            "event_id": 2226072,
            "device_id": 94,
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-09-28T12:37:37.993",
            "raw_data": "$G,5,001EC0828146,R,18030,-58,-12,69B2,81100547,13,B,1,0,0,1,98,*22"
        },
        {
            "tag_id": "81200200",
            "gateway_id": "001EC0828146",
            "event_id": 2226071,
            "device_id": 330,
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-09-28T12:37:37.993",
            "raw_data": "$G,5,001EC0828146,R,17714,-51,-12,69B2,81200200,9,B,1,0,0,1,9F,0A00*16"
        },
        {
            "tag_id": "00022759",
            "gateway_id": "001EC0828146",
            "event_id": 2226070,
            "device_id": 240,
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-09-28T12:37:37.993",
```

```
        "raw_data": "$G,5,001EC0828146,R,17499,-37,297,69B2,00022759,30,B,0,0,0,1,88,*35"
    },
    {
        "tag_id": "81100589",
        "gateway_id": "001EC0828146",
        "event_id": 2226068,
        "device_id": 219,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-09-28T12:37:37.993",
        "raw_data": "$G,5,001EC0828146,R,17252,-51,-334,69B2,81100589,1,B,1,0,0,1,98,*24"
    },
    {
        "tag_id": "0002274A",
        "gateway_id": "001EC0828146",
        "event_id": 2226062,
        "device_id": 391,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-09-28T12:37:35.987",
        "raw_data": "$G,5,001EC0828146,R,16019,-60,-111,69B2,0002274A,18,B,0,0,0,1,78,*66"
    },
    {
        "tag_id": "000209DE",
        "gateway_id": "001EC0828146",
        "event_id": 2226060,
        "device_id": 149,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-09-28T12:37:35.987",
        "raw_data": "$G,5,001EC0828146,R,15300,-60,173,69B2,000209DE,6,B,1,0,0,1,A9,*76"
    },
    {
        "tag_id": "81600065",
        "gateway_id": "001EC0828146",
        "event_id": 2226058,
        "device_id": 77,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-09-28T12:37:35.987",
        "raw_data": "$G,5,001EC0828146,R,14959,-55,-37,69B2,81600065,18,B,0,0,0,1,88,0603*2E"
    },
    {
        "tag_id": "81600320",
        "gateway_id": "001EC0828146",
        "event_id": 2226057,
        "device_id": 136,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-09-28T12:37:35.987",
        "raw_data": "$G,5,001EC0828146,R,14781,-35,-49,69B2,81600320,2,B,0,0,3,1,98,*14"
    },
    {
```

```
            "tag_id": "81600310",
            "gateway_id": "001EC0828146",
            "event_id": 2226050,
            "device_id": 179,
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-09-28T12:37:33.947",
            "raw_data": "$G,5,001EC0828146,R,13615,-31,-111,69B2,81600310,1,B,1,0,3,1,88,*27"
        },
        {
            "tag_id": "81100770",
            "gateway_id": "001EC0828146",
            "event_id": 2226049,
            "device_id": 532,
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-09-28T12:37:33.947",
            "raw_data": "$G,5,001EC0828146,R,13404,-35,12,69B2,81100770,0,B,0,0,1,1,84,*35"
        }
    ]
}
```

List Events for BB5

Retrieves a list of events generated by a particular BB5 unit.

URL Endpoint

/bb5s/[IMEI]/events

e.g. /bb5s/355001090607598/events

HTTP Method

GET

Search Fields

None

Ordering Fields

Field	Description
event_received_datetime_utc	Order by event date
event_id	Order by event unique identifier

Filter Fields / Options Supported

Field	Data Type	Description
event_id__gt	Long	Event id is greater than the supplied value
event_id__gte	Long	Event id is greater than or equal to the supplied value
event_id__lt	Long	Event id is less than the supplied value
event_id__lte	Long	Event id is less than or equal to the supplied value
event_received_datetime_utc__gt	Time and/or Date	Event date is greater than the supplied value
event_received_datetime_utc__gte	Time and/or Date	Event date is greater than or equal to the supplied value
event_received_datetime_utc__lt	Time and/or Date	Event date is less than the supplied value
event_received_datetime_utc__lte	Time and/or Date	Event date is less than or equal to the supplied value

Raw Format Supported?

Yes

Response Format

application/json

Expected Response Codes

HTTP Status Code	Meaning
401	You have not provided the relevant / correct security credentials in the request to the service.
403	You are not authorised to access the function.
200	The request was successful.
404	A tag matching the supplied ID was not found, or you are not entitled to view the record for that Tag.

Example Response

Each record in the "results" field will contain additional / different fields depending on the type of tag generating the data. Examples of the different types of Tags can be found in Appendix A.


```
{
    "total_record_count": 197,
    "filtered_record_count": null,
    "options": {
        "ordering": null,
        "filtering": [],
        "search": null,
        "skip": 0,
        "take": 10,
        "format": "Formatted"
    },
    "results": [
        {
            "tag_id": "81200076",
            "gateway_id": "001EC082B4FA",
            "voltage": 3.3000,
            "temperature": "+24 to +26",
            "uart": "X 814 Y -1908 Z 1499",
            "packet_sequence_number": 36,
            "rssi": "-65",
            "extended_data": "0A005820383134205920 2D31393038205A2031343939",
            "gps": null,
            "air_quality": null,
            "temperature_log": null,
            "bin": null,
            "event_id": 3474302,
            "device_id": 5534,
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-08-11T20:15:54.992",
            "raw_data": null
        },
        {
            "tag_id": "81200076",
            "gateway_id": "001EC082B4FA",
            "voltage": 3.3000,
            "temperature": "+21 to +23",
```

```
        "uart": "X 802 Y -1878 Z 1482",
        "packet_sequence_number": 41,
        "rssi": "-62",
        "extended_data": "0A0058203830322059202D31383738205A2031343832",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3473945,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-11T14:10:16.776",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+18 to +20",
        "uart": "X 760 Y -1839 Z 1487",
        "packet_sequence_number": 8,
        "rssi": "-62",
        "extended_data": "0A0058203736302059202D31383339205A2031343837",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3471104,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-09T13:37:55.41",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+18 to +20",
        "uart": "X 747 Y -1812 Z 1487",
        "packet_sequence_number": 49,
        "rssi": "-59",
        "extended_data": "0A0058203734372059202D31383132205A2031343837",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3464412,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-04T19:13:47.346",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
```

```
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+15 to +17",
        "uart": "X 738 Y -1818 Z 1479",
        "packet_sequence_number": 19,
        "rssi": "-65",
        "extended_data": "0A0058203733382059202D31383138205A2031343739",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3464175,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-04T15:11:11.792",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+15 to +17",
        "uart": "X 743 Y -1826 Z 1475",
        "packet_sequence_number": 23,
        "rssi": "-61",
        "extended_data": "0A0058203734332059202D31383236205A2031343735",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3464001,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-04T12:12:52.145",
        "raw_data": null
    },
    {
        "tag_id": "81200076",
        "gateway_id": "001EC082B4FA",
        "voltage": 3.3000,
        "temperature": "+18 to +20",
        "uart": "X 761 Y -1842 Z 1483",
        "packet_sequence_number": 1,
        "rssi": "-73",
        "extended_data": "0A0058203736312059202D31383432205A2031343833",
        "gps": null,
        "air_quality": null,
        "temperature_log": null,
        "bin": null,
        "event_id": 3461641,
        "device_id": 5534,
        "event_type": "Tag",
        "event_received_datetime_utc": "2020-08-02T19:52:59.776",
        "raw_data": null
```

```
},
{
    "tag_id": "81200076",
    "gateway_id": "001EC082B4FA",
    "voltage": 3.3000,
    "temperature": "+21 to +23",
    "uart": "X 969 Y -1863 Z 1340",
    "packet_sequence_number": 5,
    "rssi": "-53",
    "extended_data": "0A0058203936392059202D31383633205A2031333430",
    "gps": null,
    "air_quality": null,
    "temperature_log": null,
    "bin": null,
    "event_id": 3440082,
    "device_id": 5534,
    "event_type": "Tag",
    "event_received_datetime_utc": "2020-07-29T16:23:26.803",
    "raw_data": null
},
{
    "tag_id": "81200076",
    "gateway_id": "001EC082B4FA",
    "voltage": 3.3000,
    "temperature": "+18 to +20",
    "uart": "X 949 Y -1857 Z 1341",
    "packet_sequence_number": 1,
    "rssi": "-58",
    "extended_data": "0A0058203934392059202D31383537205A2031333431",
    "gps": null,
    "air_quality": null,
    "temperature_log": null,
    "bin": null,
    "event_id": 3434811,
    "device_id": 5534,
    "event_type": "Tag",
    "event_received_datetime_utc": "2020-07-25T20:25:08.763",
    "raw_data": null
},
{
    "tag_id": "81200076",
    "gateway_id": "001EC082B4FA",
    "voltage": 3.3000,
    "temperature": "+21 to +23",
    "uart": "X 957 Y -1860 Z 1343",
    "packet_sequence_number": 25,
    "rssi": "-57",
    "extended_data": "0A0058203935372059202D31383630205A2031333433",
    "gps": null,
    "air_quality": null,
    "temperature_log": null,
    "bin": null,
    "event_id": 3434492,
    "device_id": 5534,
```

```
            "event_type": "Tag",
            "event_received_datetime_utc": "2020-07-25T14:58:57.847",
            "raw_data": null
        }
    ]
}
```


Raw Response Example

Appendix A – Tag Types and Event Examples

The following section details the types of Tags supported by the API, and the format of data they generate.

Common Data Fields

For all tag types, there are a number of common fields that are always output. Additional fields are output depending on the type of tag and sensors attached. An example of common data is as follows:


```
{
  "tag_id": "80200133",
  "gateway_id": "5410ECF3FAC2",
  "voltage": 3.5000,
  "temperature": "+24 to +26",
  "packet_sequence_number": 7,
  "rssi": "-81",
  "alert_1": false,
  "alert_2": false,
  "extended_data": "0A0063322C32372E322C32302E33",
  "has_extended_data": true,
  "event_id": 270196679,
  "device_id": 3710,
  "event_type": "Tag",
  "event_received_datetime_utc": "2021-06-09T11:14:22.291"
}
```


Bolus

Example Event Data


```
{
  "tag_id": "000225DF",
  "gateway_id": "001EC0827CD4",
  "voltage": 3.6000,
  "temperature": "+21 to +23",
  "packet_sequence_number": 228,
  "rssi": "-118",
  "pulse": false,
  "extended_data": "0401E400E600E600E600E600E600E6",
  "has_extended_data": true,
  "temperature_log": [
    {
      "temperature_celsius": 23.0000
    },
    {
      "temperature_celsius": 23.0000
    },
    {
      "temperature_celsius": 23.0000
    },
    {
      "temperature_celsius": 23.0000
    },
    {
```

```
      "temperature_celsius": 23.0000
    },
    {
      "temperature_celsius": 23.0000
    }
  ],
  "event_id": 6996,
  "device_id": 439,
  "event_type": "Tag",
  "event_received_datetime_utc": "2020-06-25T09:37:44.322"
}
```


Contact (e.g. Window / Defibrillator)

Example Event Data


```
{
  "tag_id": "80200409",
  "gateway_id": "001EC08266B3",
  "voltage": 3.3000,
  "temperature": "+24 to +26",
  "packet_sequence_number": 20,
  "rssi": "-89",
  "extended_data": "06A2",
  "has_extended_data": true,
  "contact": {
    "circuit_1_open": true,
    "circuit_2_open": false
  },
  "event_id": 1072,
  "device_id": 9,
  "event_type": "Tag",
  "event_received_datetime_utc": "2020-09-25T11:08:09.694"
}
```


GPS

Example Event Data


```
{
  "tag_id": "81400137",
  "gateway_id": "5410ECF3FC24",
  "voltage": 3.5000,
  "temperature": "+15 to +17",
  "packet_sequence_number": 20,
  "rssi": "-118",
  "pulse": false,
  "extended_data": "010D505044731001470098",
  "has_extended_data": true,
  "gps": {
    "latitude": 50.8333,
    "longitude": -1.7833,
    "earth_quadrant": "NW"
  },
  "event_id": 3792033,
  "device_id": 389,
  "event_type": "Tag",
  "event_received_datetime_utc": "2020-08-18T09:37:39.813"
}
```

UART Tags

UART tag types output an additional raw data field, called “uart”. This field is parsed an output into further fields specific to the type of tag.

Pulse Only

For all UART tag types, Tags will periodically send pulse data feeds which do not contain any additional data in the “uart” field and are to be treated effectively as heartbeats. These can be identified by the “pulse” field being equal to true.

Example Event Data


```
{
  "tag_id": "81200093",
  "gateway_id": "001EC08266B3",
  "voltage": 3.3000,
  "temperature": "+18 to +20",
  "pin_state": "Low",
  "pulse": true,
  "packet_sequence_number": 22,
  "rssi": "-100",
  "extended_data": "0A00",
  "has_extended_data": true,
  "event_id": 3481632,
  "device_id": 6285,
  "event_type": "Tag",
  "event_received_datetime_utc": "2020-09-07T09:50:56.272"
}
```


Legionella

Example Event Data


```
{
  "tag_id": "80200408",
  "gateway_id": "001EC08266B3",
  "voltage": 3.3000,
  "temperature": "+24 to +26",
  "pin_state": "Low",
  "uart": "23.53,23.53,5.51",
  "pulse": false,
  "packet_sequence_number": 20,
  "rssi": "-89",
  "extended_data": "0A0032332E35332C32332E35332C352E3531",
  "has_extended_data": true,
  "temperature_readings": {
    "hot_temperature_celsius": 23.5300,
    "cold_temperature_celsius": 23.5300
  },
  "power": {
    "battery_volts": 5.5100
  },
  "event_id": 3502996,
  "device_id": 6289,
  "event_type": "Tag",
  "event_received_datetime_utc": "2020-09-22T14:37:15.974"
}
```

Pipe Sensor

Example Event Data


```
{
  "tag_id": "81200093",
  "gateway_id": "001EC08266B3",
  "voltage": 3.3000,
  "temperature": "+18 to +20",
  "pin_state": "Low",
  "uart": "c2,21.7,21.1",
  "pulse": false,
  "packet_sequence_number": 22,
  "rssi": "-100",
  "extended_data": "0A0063322C32312E372C32312E31",
  "has_extended_data": true,
  "temperature_readings": {
    "hot_temperature_celsius": 21.7000,
    "cold_temperature_celsius": 21.1000
  },
  "event_id": 3481594,
  "device_id": 6285,
  "event_type": "Tag",
  "event_received_datetime_utc": "2020-09-07T08:14:48.718"
}
```


Bin / Depth Sensor

Example Event Data


```
{
  "tag_id": "80200299",
  "gateway_id": "001EC0827CD4",
  "voltage": 3.5000,
  "temperature": "+15 to +17",
  "pin_state": "Low",
  "uart": "d,31",
  "packet_sequence_number": 21,
  "rssi": "-88",
  "pulse": false,
  "extended_data": "0A00642C3331",
  "has_extended_data": true,
  "bin": {
    "depth_cm": 31.0000
  },
  "event_id": 264864,
  "device_id": 97,
  "event_type": "Tag",
  "event_received_datetime_utc": "2020-07-06T12:40:04.882"
}
```


Air Quality

Air Quality sensors send data packets in cycles of 4; each cycle contains a different set of readings.

Example Event Data

Cycle 1


```
{
  "tag_id": "81200198",
  "gateway_id": "001EC082D54C",
```

```
  "voltage": 4.3000,
  "temperature": "+18 to +20",
  "pin_state": "Low",
  "uart": "p,1,195,65,0,0",
  "packet_sequence_number": 0,
  "rssi": "-78",
  "pulse": false,
  "extended_data": "0A00702C312C3139352C36352C302C30",
  "has_extended_data": true,
  "air_quality": {
    "pm25_standard": 1.0000,
    "particles_03um": 195.0000,
    "particles_05um": 195.0000,
    "particles_10um": 195.0000,
    "particles_25um": 195.0000
  },
  "event_id": 319636,
  "device_id": 534,
  "event_type": "Tag",
  "event_received_datetime_utc": "2020-07-07T11:24:16.242"
}
```


Cycle 2


```
{
  "tag_id": "81200198",
  "gateway_id": "001EC082D54C",
  "voltage": 4.3000,
  "temperature": "+18 to +20",
  "pin_state": "Low",
  "uart": "t,23.70,35.40",
  "packet_sequence_number": 1,
  "rssi": "-75",
  "pulse": false,
  "extended_data": "0A00742C32332E37302C33352E3430",
  "has_extended_data": true,
  "air_quality": {
    "temperature_degrees_c": 23.7000,
    "humidity_percentage": 35.4000
  },
  "event_id": 319646,
  "device_id": 534,
  "event_type": "Tag",
  "event_received_datetime_utc": "2020-07-07T11:24:23.643"
}
```


Cycle 3


```
{
  "tag_id": "81200198",
  "gateway_id": "001EC082D54C",
  "voltage": 4.3000,
  "temperature": "+18 to +20",
  "pin_state": "Low",
  "uart": "g1,6.2,42,0.69,16059",
  "packet_sequence_number": 2,
  "rssi": "-75",
  "pulse": false,
  "extended_data": "0A0067312C362E322C34322C302E36392C3136303539",
  "has_extended_data": true,
  "air_quality": {
    "nH3_ppm": 6.2000,
```

```
    "cO_ppm": 42.0000,
    "nO2_ppm": 0.6900,
    "c3H8_ppm": 16059.0000
  },
  "event_id": 319656,
  "device_id": 534,
  "event_type": "Tag",
  "event_received_datetime_utc": "2020-07-07T11:24:30.722"
}
```


Cycle 4


```
{
  "tag_id": "81200195",
  "gateway_id": "001EC082D54C",
  "voltage": 4.3000,
  "temperature": "+24 to +26",
  "pin_state": "Low",
  "uart": "g2,5128,99999,25,34",
  "packet_sequence_number": 32,
  "rssi": "-15",
  "pulse": false,
  "extended_data": "0A0067322C353132382C39393939392C32352C3334",
  "has_extended_data": true,
  "air_quality": {
    "c4H10_ppm": 5128.0000,
    "cH4_ppm": 99999.0000,
    "h2_ppm": 25.0000,
    "c2H5OH_ppm": 34.0000
  },
  "event_id": 319669,
  "device_id": 532,
  "event_type": "Tag",
  "event_received_datetime_utc": "2020-07-07T11:24:39.752"
}
```


Electricity Contact Pulse Meter Readers

These types of meters provide data regarding electricity usage in pulse counts, as well as whether or not the power supply is switched on, and also whether a power cable is currently connected to the unit.

Example Event Data


```
{
  "tag_id": "80200267",
  "gateway_id": "5410ECF41BA6",
  "voltage": 3.6000,
  "temperature": "+18 to +20",
  "pin_state": "Low",
  "uart": "SB,1,0,0,0,0,0,26,0",
  "pulse": false,
  "packet_sequence_number": 6,
  "rssi": "-64",
  "alert_1": false,
  "alert_2": false,
  "extended_data": "0A0053422C312C302C302C302C302C302C32362C30",
  "has_extended_data": true,
  "meter": {
    "readings": [
      {
```

```
        "number": 1,
        "reading": 26,
        "description": "Pulse Count"
      },
      {
        "number": 2,
        "reading": 0,
        "description": "Pulse Count"
      }
    ],
    "update_reason": {
      "id": 0,
      "description": "Normal Periodic"
    },
    "shutdown_reason": {
      "id": 0,
      "description": "None"
    },
    "power_switch_status": {
      "id": 1,
      "description": "Powered"
    },
    "contact_status": {
      "id": 0,
      "description": "Connected"
    }
  },
  "event_id": 270246945,
  "device_id": 4753,
  "event_type": "Tag",
  "event_received_datetime_utc": "2021-06-09T13:26:41.406"
}
```


Sound

Example Event Data


```
{
  "tag_id": "80200407",
  "gateway_id": "001EC08266B3",
  "voltage": 3.3000,
  "temperature": "+24 to +26",
  "pin_state": "Low",
  "uart": "s1,47.0,40.6,62.0",
  "pulse": false,
  "packet_sequence_number": 20,
  "rssi": "-89",
  "extended_data": "0A0073312C34372E302C34302E362C36322E30",
  "has_extended_data": true,
  "sound": {
    "average_db": 47.0000,
    "minimum_db": 40.6000,
    "maximum_db": 62.0000
  },
  "event_id": 3494621,
  "device_id": 6287,
  "event_type": "Tag",
  "event_received_datetime_utc": "2020-09-16T15:29:08.82"
}
```