fieldin

Fieldin
Trackings API

2

V2.0.0

General

This API provides an endpoint to get samples (trackings) form the installed units (logger).

It is important to understand that the time passed to the API is the time of the sample arriving to the DB and not the time it was taken in the field.
Due to cellular coverage issues, there are cases when a lag of a couple of hours may occur - providing the time in DB allow querying the API sequencily with no need to recheck the same time over and over without knowing if all the data arrived. The API will return samples with the real timestamp the sample was taken.
I.E, querying the API every 24 hours, will get old data samples that might have arrived even in a dealy of 48 hours.

Notice:

1. ALL requests should be made with GET method
2. ALL requests should have 'accept: application/json' header

Discovery

In order to discover which device has new data, you should use the discovery API. The discovery API will return all devices that received new data (in DB) in the requested period.

https://api.fieldintech.com/external/2.0.0/trackings/discover?api-key={api_key}&start_time_utc={start_time_utc}&end_time_utc={end_time_utc}

Parameters

Name	Description	Mandatory
start_time_utc	The date to bring data from I.E: 2017-1-20	Yes
end_time_utc	The date to bring data to I.E: 2017-1-21	Yes

Response

Name	Type	Description
info	array	Meta data for request include: start_time_utc, end_time_utc, devicecount
info.start_time_utc	string	Request limit start time
info.end_time_utc	string	Request limit end time
info.devicecount	int	Total devices found
data	array	Devices data
data.device_id	int	The device ID
data.device_alias	string	Device Alias
data.device_code	string	Device code (as appears on the device box on the sprayer/tractor)
data.device_type	string	Type of Device (tractor,hedger,shaker, etc)
data.raw_data_endpoint	string	URL link to trackings/export
data.shakes_endpoint	string	Only if device_type shaker. URL link to trackings/shaking

Response example

```
{
  "info": {
      "start_time_utc": "2019-11-05",
      "end_time_utc": "2019-11-06",
      "devicecount": 103
  },
  "data": [
      {
      "device_id": 260,
      "device_alias": "25-3489 (AOF7)",
      "device_code": "BG93997 (old)",
      "device_type": "sprayer",
      "raw_data_endpoint":
"https://api.fieldintech.com:443/external/2.0.0/trackings/export?api-ke
y=3bd19186cabc36e44ae7&start_time_utc=2019-11-05&end_time_utc=2019-11-0
6&device_id=260"
      }
  ]
}
```

Trackings

Use this endpoint to bring all the trackings for your account between certain dates as explained above..

The data is ordered by timestamp and served in GEOJSON format.

Endpoint

https://api.fieldintech.com/external/2.0.0/trackings/export?api-key={api_key}&device_id={device_id1,device_id2,device_id3}&start_time_utc={start_time_utc}&end_time_utc={end_time_utc}

Parameters

Name	Description	Mandatory
device_id	device_id	Yes
start_time_utc	The date to bring data from I.E: 2017-1-20	Yes
end_time_utc	The date to bring data to I.E: 2017-1-21	Yes

Response

Response

Name	type	Description
id	int	Device Id
device_alias	string	Device Alias
timestamp	int	Action time in UNIX timestamp
ms	int	Action duration in miliseconds
headings	string	Device heading

speed	string	Device speed in the relevant units
speed_units	string	Device speed units
rpm	int	Revolutions per minute of the device's sample
action	int	The device's sample action code (see **action code description** for more details)
eventcode	int	The device's sample event code (see **event code description** for more details)
pto	int	A flag indicating whether the device's pto is on
right_boom	int	A flag indicating whether the device is spraying right
left_boom	int	A flag indicating whether the device is spraying left

Example

https://api.fieldintech.com/external/2.0.0/trackings/export?api-key=1a2b3c4d5e67891&device_id=1&start_time_utc=2017-01-01 00:00:00&end_time_utc=2017-01-01 06:00:00

Action code description.

Each tracking sample has an action code attached to it

Action Code	Description
1	spraying [sprayer,tractor]
2	refill [sprayer,weed_sprayer,tractor,water_truck,mix_tank]
3	stand
4	off
5	shaking [shaker]
6	driving
7	waiting [receiver,bankout]
8	unloading [receiver,platform]
9	collecting [receiver,platform,almonds_bankout,gondola]
10	loading [bankout]
11	mowing [mower]
12	burning [burner]
13	flying [plane]
14	moving [shaker]
15	shredding [shredder]
16	hedging [hedger]
17	harvesting [harvester,vegetable_harvester,grape_harvester]
18	sweeping [sweeper,sanitation_sweeper]
19	nursing [water_truck]
20	conditioning [conditioner]
21	offloading [bankout]
22	blowing [blower,harvest_blower]
23	spreading [spreader]
24	carrying [big_fork,utility_cart]
25	working [smart_tractor,unknown,harvest_smart_tractor]
26	discing [disc]

27	weed_spraying [weed_sprayer]
28	maintenance []
29	patrolling [atv]
30	levelling [smudge_bar,landplane,laser_leveller]
31	flattening [x_plain]
32	dusting [duster]
33	tilling [eliminator]
34	scraping [scraper]
35	chiseling [chisel]
36	ripping [ripper]
37	deadhead_driving [vegetable_harvester]

Event code description.

Each tracking sample has an event code attached to it

Event Code	Description
10	Tracking - Sleeping
20	Input sensor 1 On
21	Input sensor 1 Off
23	Main Power on
24	Main Power off
27	Tracking - Time passed
28	Tracking - Distance
30	Start Trip
31	End Trip
186	Input sensor 2 On
187	Input sensor 3 On
188	Input sensor 4 On

189	Input sensor 5 On
190	Input sensor 6 On
191	Input sensor 2 Off
192	Input sensor 3 Off
193	Input sensor 4 Off
194	Input sensor 5 Off
195	Input sensor 6 Off
211	Tracking - Angle change

Response example

*speed_units will always be kmh

```json
{
    "type":"FeatureCollection",
    "features":[
        {
            "type":"Feature",
            "properties":{
                "id":1,
                "device_alias":"ExampleDevice",
                "timestamp":1496359003,
                "ms":384,
                "heading":"273.00",
                "speed":1.4912904,
                "speed_units":"kmh",
                "rpm":1678,
                "action":"spraying",
                "eventcode":10,
                "pto":0,
                "right_boom":null,
                "left_boom":null
            },
            "geometry":{
                "type":"Point",
                "coordinates":[
                    -119.212127700,
                    35.629673000
```

```
                    ]
                }
            }
        ]
    }
```

Shaking

Use this endpoint to bring all the shakes for your account between certain dates as explained above (time is based on the time samples received in the server).
The data is ordered by timestamp and served in GEOJSON format.

Endpoint

https://api.fieldintech.com/external/2.0.0/trackings/shaking?api-key={api_key}&device_id={device_id1,device_id2,device_id3}&start_time_utc={start_time_utc}&end_time_utc={end_time_utc}

Parameters

Name	Description	Mandatory
device_id	device_id	Yes
start_time_utc	The date to bring data from I.E: 2017-1-20	Yes

end_time_utc	The date to bring data to I.E: 2017-1-21	Yes

Response

Name	type	Description
alias	string	Device Alias
time_netto	decimal	The amount of time it took to shake in seconds. If there were 2 shakes (or more) it's the sum in second without the time between shakes.
time_brutto	decimal	The amount of time it took from the end of shake of the previous shake, to the end of the current shake (includes the driving time to the tree).
shakes_count	decimal	Number of shakes
headings	decimal	Device heading
date	string	UTC day
timestamp	string	Shake time in UNIX timestamp

Example

https://api.fieldintech.com/external/2.0.0/trackings/shaking?api-key=1a2b3c4d5e67891&device_id=1&start_time_utc=2017-01-01 00:00:00&end_time_utc=2017-01-01 06:00:00

Response example

```json
{
    "type":"FeatureCollection",
    "features":[
        {
            "type":"Feature",
            "properties":{
                "time_netto":2.50200000000000000000,
                "time_brutto":3.00000000000000000000,
                "shakes_count":1.00000000000000000000,
                "headings":"0.00",
                "alias":"1234",
                "date":"2019-09-30",
                "timestamp":"1569850672"
            },
            "geometry":{
                "type":"Point",
                "coordinates":[
                    -119.212127700,
                    35.629673000
                ]
            }
        }
    ]
}
```