Getting Started - SignalFire Cloud API

Rev 1.2

What is an API?

An API (Application Programming Interface) provides a way for others to access the data managed by SignalFire. All data transfer is done using standard HTTP and REST protocols.

Here we provide methods that access the data one would be most likely to request including:

- User/Group info about your SF Cloud account
- List of Ranger nodes you own
- All metrics measured on any given Ranger node
- Tag history for a ranger

How would you use this API?

There are 3 very common languages used to access APIs: Python, JavaScript, and curl. Functionally, it can be done in any language. Below are some examples.

Python 3

```
import requests
response = requests.get('https://api.signal-fire.cloud/api/v1/sfcloud/group/me/',
                        headers={'Authorization': 'Token YOUR_API_ACCESS_TOKEN'})
data = response.json()
```

JavaScript

```
fetch('https://api.signal-fire.cloud/api/v1/sfcloud/group/me/', {
    headers: {Authorization: 'Token YOUR_API_ACCESS_TOKEN'}
})
    .then(response => response.json())
    .then(data => {
    console.log(data)
})
```

curl – Standard library (bash/Linux)

```
curl https://api.signal-fire.com/api/v1/sfcloud/group/me/ --header "Authorization: Token YOUR_API_TOKEN_HERE"
```

In this example, we will be using Python as an example on how to get started using the API for SignalFire cloud.

To get started, please ensure you have Python 3 (3.8+) installed, and the Python requests package installed. If you do not have requests installed, you can get it by running pip3 install requests

Then head over to https://api.signal-fire.cloud/api/, and login with your SFCloud username/password.
Once logged in, you can review the API schema in 3 available formats using the links at the top of the page.

NOTE: If you are unable to login, please contact SignalFire so we can set up your API account. API access is a paid-only feature, separate from base SFCloud access.

User Accounts

The API account is technically a separate account from your SignalFire Cloud account. However, the username and password to login is linked to your SF Cloud account.

Each account is given an **API token**. This token must be passed with any request that requires you to be logged in. Information on how to do so is included in this guide.

Important Links

There are a few links that you should be aware of to use the API effectively.

- `/`, `/api/`, `/api/v1/`
  - redirects to `/api/v1/schema/openapi/`
- `/api/v1/schema/openapi/`
  - OpenAPI standard YAML schema. Lists everything about the API in one standardized document available to download.
  - This also provides a login page for the API in the top right to browse while authenticated.
- `/api/v1/schema/swagger/`
  - An interactive browsable list of endpoints in the API
- `/api/v1/schema/redoc/`
  - An alternative interactable list of endpoints
- `/api/v1/auth/login/`
  - Login URL for the API

What kind of data do you receive?

At the bottom of our Swagger documentation page, you can find the models of data you can receive from the API. All data is returned in JSON format.

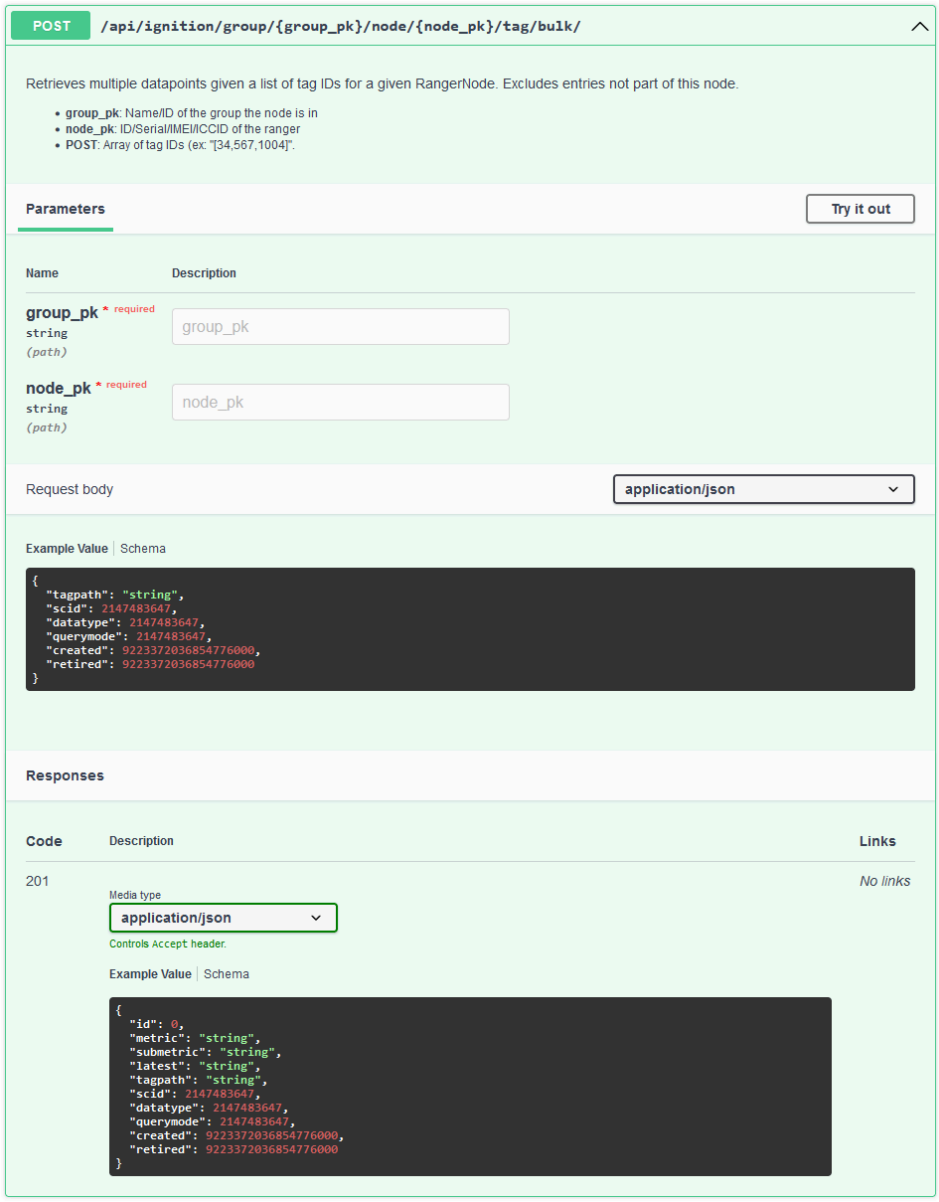
Once you have this data, you can manipulate it, sort it, filter it, and store it in any way you wish.

List of endpoints

All endpoints fall under `/api/v1/sfcloud/`. All endpoints have a trailing slash `/` that will be automatically appended with a redirect.

The list of endpoints can be found under our Swagger documentation available at /api/schema/swagger.

Each endpoint has an expandable menu that includes a description of the endpoint, the parameters to send to it, the possible return types, and a **Try It Out** button to play with the endpoints directly in the browser.



Reading and Using the Schema

We provide our API schema in 3 formats. You can view any of these using the links at the top of the page.

- **OpenAPI:** YAML representation of our entire schema. You can feed this into other API tools to utilize it.
- **Swagger:** (Recommended) provides an interactive playground of every endpoint, and descriptions for each one.
- **ReDoc:** An alternative to Swagger. Provides similar functionality, but with a different layout.

Navigate over to the Swagger documentation page to get started using the API.
Here you can click on any of the endpoints to expand it and get more information. It will list the description of what each endpoint returns, and the required variables to provide to get access to your information.

Object IDs

When navigating the API, you will be required to input the IDs of various objects to retrieve them properly. These usually have multiple options for what you can put here.

- **Group** (group_id) can be substituted with the following
  - Short name of the group (aka demo)
  - The word me (will automatically resolve to your primary group)
- **Node** (node_id) can be substituted with the following
  - ID number
  - Serial Number of the ranger
  - IMEI of the ranger
- **Tag** (tag_path) (sometimes referred to as metric_path)
  - Must be wrapped in @ symbols when using in the URL (not necessary for Swagger input, or using via POST)

Utility Endpoints

We provide some endpoints for convenience, instead of having to manually accrue additional data.
These are the ability to query all Ranger's at once, or multiple tags at once. Whenever you see the word bulk in the endpoint, that means you can submit a list of those IDs via POST in JSON format to retrieve info from multiple instances at once.

- Submit a JSON object with 2 keys, nodes and tags. Each of those keys contains an array as above.
- Ex: {"nodes": [“RA000512”], "metric_paths": [“battery”, “node info/online”]}

Fetching live data

We do not have any endpoints to fetch data in real time. The best option is to poll our API and retrieve data at a set interval.

Historical data

Currently, we only support pulling data up to 12 week spans at a time. This is a limitation we will look at adjusting in the future.

Rate Limiting

Anonymous users are allowed to make up to 50 requests per day (subject to change). This allows one to test out the API without having an account, with limited access.

Authenticated users are allowed to make up to 10,000 requests per day. Some endpoints may be more limited per day.

Tokens

The API token provided to you is what you will use to access the API via command-line. You can view your token using the “API TOKEN” link at the top of the page. Here you can also rotate your token, creating a new one and invalidating the old one.

Unless manually invalidated by either us or you, the token will never expire. It will grant full access to your API account, so treat it securely, like a password. We recommend storing it in an environment variable (or other secret location) for use in scripts.

You cannot access any URLs that involve viewing or modifying the token using the token itself. You must authenticate with username/password to deal with your token.

Examples

NOTE: This following block of code would precede all examples. (Aka paste it before each example)

```
import requests
import os
API_TOKEN = os.getenv('API_TOKEN')
headers = {'Authorization': f'Token {API_TOKEN}'}
```

User and group info

```
response = requests.get('https://api.signal-fire.cloud/api/v1/sfcloud/group/me/', headers=headers)
data = response.json()
print(data)
```

Get all Rangers in your group

```
response = requests.get('https://api.signal-fire.cloud/api/v1/sfcloud/group/me/node/',
                        headers=headers)
data = response.json()
print(data)
```

Get all possible tags on a Ranger

```
# Note: Instead of Serial Number, you can also use the IMEI of your Ranger, or the database ID value
response = requests.get('https://api.signal-fire.cloud/api/v1/sfcloud/group/me/node/RA001395/tag/',
                        headers=headers)
# RA001395 refers to the Serial# of the Ranger, retrieved from previous example
data = response.json()
print(data)
```

Get measurement history for a given tag

```
# Note: response is returned in pages
response = requests.get('https://api.signalfire.cloud/api/v1/sfcloud/group/me/node/RA001395/tag/@battery@/history/',
                        headers=headers)
data = response.json()
print(data)
```

Get a certain page of history

```
response =
requests.get('https://api.signalfire.cloud/api/v1/sfcloud/group/me/node/RA001395/tag/@battery@/data/?offset=300&limit=20,
             headers=headers)
data = response.json()
print(data)
```

Get a data point by timestamp

```
response =
requests.get('https://api.signalfire.cloud/api/v1/sfcloud/group/me/node/RA001395/tag/@battery@/data/1660600308804/',
             headers=headers)
data = response.json()
print(data)
```

Get latest value for all data points on a Ranger

```
response = requests.get('https://api.signal-fire.cloud/api/v1/sfcloud/group/me/node/RA001395/tag/all/',
                        headers=headers)
data = response.json()
print(data)
```

Get latest value for specific data points

```
response = requests.post('https://api.signal-fire.cloud/api/v1/sfcloud/group/me/node/RA001395/tag/bulk/latest/',
                         headers=headers,
                         json=['battery', 'node info/online'])
data = response.json()
print(data)
```

Get online status of all Rangers

```
response = requests.get('https://devapi.signal-fire.cloud/api/v1/sfcloud/group/me/node/all/tag/@node
info/online@/latest/',
                        headers=headers)
data = response.json()
print(data)
```

Get Data between 2 dates

```
response = requests.get('https://api.signal-
fire.cloud/api/v1/sfcloud/group/me/node/RA001395/tag/@battery@/between/1676645733537/1676648730646/',
             headers=headers)
data = response.json()
print(data)
```

Common patterns

This section provides a bit more info about common use-cases for this kind of API.

Getting tag path to fetch data

This is probably the most common use case for this API.
Normally, you will make multiple API calls to fetch the data you require. Not all of them will be required if you already know the info.

```
# First, we get the Ranger noes we have access to and pick one from the list
response = requests.get('https://api.signal-fire.cloud/api/v1/sfcloud/group/me/node/',
                        headers=headers)
nodes = response.json()


# Next, we get the list of tags on the Ranger so we know what to grab the value for
response = requests.get('https://api.signal-fire.cloud/api/v1/sfcloud/group/me/node/RA001395/tag/',
                        headers=headers)
tags = response.json()


# Finally, we can grab the latest value for a tag on a given Ranger node
response = requests.get('https://api.signal-fire.cloud/api/v1/sfcloud/group/me/node/RA001395/tag/@battery@/data/latest/',
             headers=headers)
data = response.json()


# Or we can grab the latest value for all the tags on the Ranger
response = requests.get('https://api.signal-fire.cloud/api/v1/sfcloud/group/me/node/RA001395/tag/all/data/latest/',
                        headers=headers)
all_data = response.json()


# Or maybe the last 24 hours of a pump
response = requests.get('https://api.signal-
fire.cloud/api/v1/sfcloud/group/me/node/RA001395/tag/@dev1/ain1@/data/last24hours/',
             headers=headers)
all_data = response.json()
```

Reference

Tag datatype

Type	Datatype
0	Integer / Long
1	Float / Double
2	String
3	Date

Quality Codes

Code	Meaning
192	GOOD
500	BAD_STALE