



Sensecap Openapi 文档

SenseCAP API 简介

简介	2
----------	---

HTTP API

HTTP API 快速入门	3
HTTP API 访问指南	5

HTTP API 方法说明

组管理API	8
设备管理API.....	14
设备数据API.....	30

Data OpenStream API

Data OpenStream API快速入门.....	38
Data OpenStream API方法说明.....	41

附录

传感器的类型表	43
测量值信息表	45

SenseCAP API 简介



SenseCAP是一套工业级传感网络系统，实现远距离和超低功耗的环境物理量数据采集，包含可靠易用的硬件产品和数据API服务。详细产品信息可以查看[solution.seeed.cc \(page 0\)](http://solution.seeed.cc (page 0))

SenseCAP提供易用的云平台，用户可以通过手机App扫描二维码将设备绑定到对应账户，并在云平台上管理设备和查看传感器的数据，SenseCAP云平台提供一套API，用户可以从云平台上获取数据并做二次集成开发。

SenseCAP API是专门用于用户做物联网设备管理和数据调用的接口，它由基于HTTP API和Data OpenStream的API组成

- 基于HTTP API，用户可以管理LoRa和NB-IoT设备，获取原始数据或者历史数据。
- 基于Data OpenStream API，用户可以实时监控传感器的测量数据。

HTTP API 快速入门

Summary:

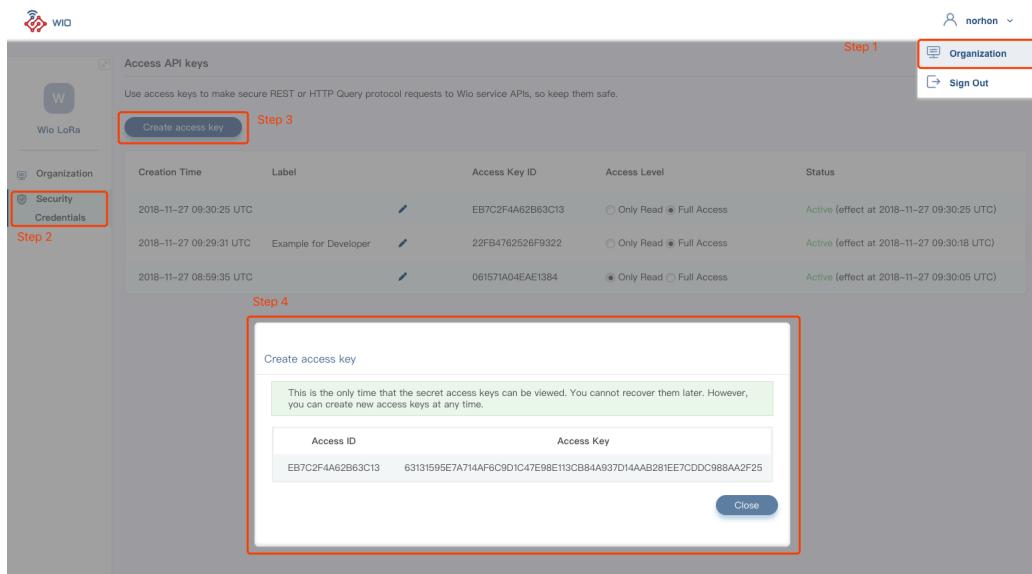
本小节我们将向您展示如何使用SenseCAP HTTP API进行第一次HTTP API调用。

准备工作

请先向SenseCAP销售那里获得您的SenseCAP帐户，并确保您可以登录SenseCAP云平台 <https://sensecap.seeed.cc>。

获取Access Key

1. 登录SenseCAP云平台：<https://sensecap.seeed.cc>
2. 前往“Organization/Security Credentials”。
3. 点击“Create access key”。
4. 获取“Access ID”和“Access Key”。



获取所有账号下已绑定设备的EUI（设备ID）

使用curl发出HTTP请求。下面的示例将调用API来获取属于您的所有设备的EUI。

```
curl --user "<access_id>:<access_key>" \
https://sensecap-openapi.seeed.cc/1.0/lists/devices/eui
```

请将您刚获取到的Access ID和Access Key将上面的和替换掉。然后将会得到和下面一样格式的输出结果：

```
{
  "code": "0",
  "data": {
    "gateway": [
      "2CF7F1121130003F",
      "2CF7F1100470001A"
    ],
    "node": [
      "2CF7F12210400031",
      "2CF7F12210400005",
      "2CF7F12210400023"
    ]
  },
  "time": 0.314
}
```

HTTP API 访问指南

HTTP方法

SenseCAP API 使用五种HTTP方法来访问：

- GET
 - 发出GET请求以获取数据。GET请求永远不会更新或更改您的数据，因为它们是安全且幂等的。
- POST
 - 使用POST请求创建新资源。例如，向设备组发出创建新组的POST请求。
- PATCH
 - 发出PATCH请求更新资源。对于PATCH请求，您只需要提供需要更改的数据即可。
- PUT
 - 使用PUT请求创建或更新资源。这对于同步订阅服务器数据非常有用。
- DELETE
 - 发出DELETE请求以删除资源。

❶ Note: 如果您的运营商不允许GET或POST以外的HTTP操作，请使用HTTP Method tunneling —— 一种通过POST方法隧道传输任何HTTP方法的机制，但需要在X-HTTP-Method-Override头文件中包含您打算使用的方法。

HTTP 的请求和响应

使用HTTP基本身份验证对请求进行身份验证。

HTTP基本身份验证

HTTP基本身份验证是RESTfull API身份验证最常见的方法之一。我们使用Access ID作为用户名，Access Key作为密码。各种HTTP客户端库基本都会支持基本身份验证，在本文档中，我们使用curl通过--user选项来指定基本身份验证凭据。

您可以通过[SenseCAP云平台](#) 来创建Access Key。请参阅HTTP API 快速入门，了解如何获取访问密钥。

API 响应

所有响应字段都遵循小写和下划线约定

字符串响应成功

```
{  
    "code": "0",  
    "data": "  
        // string  
    "  
}
```

对象响应成功

```
{  
    "code": "0",  
    "data": {  
        // object  
    }  
}
```

数组响应成功

```
{  
    "code": "0",  
    "data": [  
        // Array  
    ]  
}
```

响应错误

```
{  
    "code": "1001",  
    "msg": "error message"  
}
```

组管理API

创建新设备组

POST {host}/1.0/group

此功能用于创建新设备组，上限为50组。

Body Parameters

name (required) string 自定义组名称

返回示例

```
{  
    "code": "0",  
    "data": {  
        "name": "custom group name",  
        "unique_name": "39AE810FF660B42F"  
    }  
}
```

请求示例

```
curl --request POST \  
--url '{host}/1.0/group' \  
--user '<username>:<password>' \  
--header 'content-type: application/x-www-form-urlencoded' \  
--data '{"name":"device group name"}' \  
--include
```

更新设备组信息

PATCH {host}/1.0/group/:unique_name

更新组的名称

Path Parameters

unique_name (required) string 设备组的唯一标识

Body Parameters

name (required) string 设备组的新名称

返回示例

```
{  
    "code": "0",  
    "data": ""  
}
```

请求示例

```
curl --request PATCH \  
      --url '{host}/1.0/group/:unique_name' \  
      --user '<username>:<password>' \  
      --header 'content-type: application/x-www-form-urlencoded' \  
      --data '{"name":"new name of the device group"}' \  
      --include
```

删除设备组

DELETE {host}/1.0/group/:unique_name

删除一个设备组，删除后该组内的设备将自动移到默认组中

Path Parameters

unique_name (required) string 设备组的唯一标识

返回示例

```
{  
    "code": "0",  
    "data": ""  
}
```

请求示例

```
curl --request DELETE \  
      --url '{host}/1.0/group/:unique_name' \  
      --user '<username>:<password>' \  
      --header 'content-type: application/json' \  
      --include
```

获取设备组列表

DELETE {host}/1.0/lists/group

获取所有设备组

返回示例

```
{  
    "code": "0",  
    "data": [  
        {  
            "name": "group1",  
            "group_unique_name": "220BD4AD87EB6B68",  
            "dev_cnt": "3",  
            "online_cnt": "1",  
            "created": "1563947209000"  
        }  
    ]  
}
```

请求示例

```
curl --request GET \
--url '{host}/1.0/lists/group' \
--user '<username>:<password>' \
--include
```

获取设备组和组内的设备列表

DELETE {host}/1.0/lists/group/devices

获取所有的组以及每个组内的设备。

其中有一个默认组是虚拟组，他的group_unique_name为空。所有的设备绑定时如果未指定组，将会默认分配到默认组中，之后可以用户可以自由分配设备到其他组中。

返回示例

```
{  
    "code": "0",  
    "data": {  
        "node": [  
            {  
                "group_name": "Default",  
                "group_unique_name": "",  
                "devices": []  
            },  
            {  
                "group_name": "device group name",  
                "group_unique_name": "DAF6EECDCDCA6BB2",  
                "devices": [  
                    {  
                        "dev_eui": "2CF7F12004100005",  
                        "dev_name": "device name of LoRa node",  
                        "lon": "113.32134",  
                        "lat": "22.4523",  
                        "online_status": "1", // "0"表示电池低电量，“1”表示  
                        电池电量正常  
                        "battery_status": "1" // "0"表示设备离线，“1”表示设  
                        备在线  
                    }  
                ]  
            }  
        ]  
    }  
}
```

请求示例

```
curl --request GET \  
    --url '{host}/1.0/lists/group/devices' \  
    --user '<username>:<password>' \  
    --header 'content-type: application/json' \  
    --include
```

移动设备到其它组

DELETE

{host}/1.0/group/:unique_name/devices

Path Parameters

unique_name (required) string 要移动到的组的unique_name

Body Parameters

eui (required) string 设备唯一标识

返回示例

```
{  
  "code": "0",  
  "data": ""  
}
```

设备管理API

获取网关列表

GET {host}/1.0/lists/gateway

获取网关的列表信息

Query Parameters

offset number 当前页码，从1开始，默认为1

length string 每页获取的数量，默认为20

```
// response
{
    "code": "0",
    "data": {
        "page": {
            "count": "8"
        },
        "list": [
            {
                "dev_eui": "2CF7F1120410001D",
                // gateway EUI, equipment unique identity
                "dev_name": "gtw seeed 2018csc",
                // gateway name
                "dev_area": "Asia",
                // gateway area, Europe America Asia
                "dev_country": "CHN",
                // name country, defined by customer
                "online_status": "0",
                // 1 online, 0 offline
                "description": "organization first gateway",
                // desc something
                "activate_time": "1543990216000",
                // when the customer activates the device, unit milliseconds
                "latestmsg_time": "1543990216000",
                // the latest message time, unit milliseconds
                "production_time": "1539747273000",
                // the gateway production time, unit milliseconds
                "frequency": "CN_470_510",
                // frequency
                "position": {
                    // geographic, set by customer
                    "lat": "22.2235",
                    // geographic.latitude
                    "lon": "113.21",
                    // geographic.longitude

                    "ver_hardware": "1.0.3",
                    // hardware version
                    "ver_software": "1.0.2",
                    // software version
                    "antenna_placement": "indoor",
                    // antenna placement, set by customer
                }
            ]
        }
    }
}
```

{}

请求示例

```
curl --request GET \
--url {host}/1.0/lists/gateway?offset=1&length=2 \
--user '<username>:<password>'
```

获取网关详情

GET {host}/1.0/gateway/:gateway_eui

获取指定网关的详情信息

Path Parameters

gateway_eui (required) string 设备唯一标识

```
// response
{
    "code": "0",
    "data": {
        "dev_eui": "2CF7F1120410001D", // gateway EUI, equipment unique identity
        "dev_name": "gtw seeed 2018csc", // gateway name
        "dev_area": "Asia", // gateway area, Europe America Asia
        "dev_country": "CHN", // name country, defined by customer
        "group_name": "Group for Dev", // the group name, if customer set a group for the device
        "access_read": "68B04B2BD1AE8367A4B501BF1C478024DC901A18657AAEBEB4C07AE9B10A0B15", // device access code for read
        "access_write": "D978F7DF08B3C933CB6C0A23F303F1460572E60348E240C8432FB482FC11A7A3", // device access code for write
        "online_status": "0", // 1 online, 0 offline
        "description": "organization first gateway", // desc something
        "activate_time": "1543990216000", // when the customer activates the device, unit milliseconds
        "latestmsg_time": "1543990216000", // the last message time, unit milliseconds
        "production_time": "1539747273000", // the gateway production time, unit milliseconds
        "frequency": "CN_470_510", // frequency
        "position": { // geographic, set by customer
            "lat": "22.2235", // geographic.latitude
            "lon": "113.21" // geographic.longitude
        },
        "ver_hardware": "1.0.3", // hardware version
        "ver_software": "1.0.2", // software version
        "antenna_placement": "indoor", // antenna placement, set by customer
    }
}
```

请求示例

```
curl --request GET \
--url {host}/1.0/gateway/:gateway_eui \
--user '<username>:<password>'
```

更新网关详情信息

PATCH {host}/1.0/gateway/:gateway_eui

更新网关的详情信息，以下Body Parameters需选填任意一个或多个

Path Parameters

gateway_eui (required) string 设备唯一标识

Body Parameters

group_unique_name string 设备组的唯一标识

dev_name string 设备名称

dev_country string ISO 3166-1 国家/地区代码,
例如代码“CHN”表示中国

position object
{ "lon": "longitude", "lat": "latitude", "alt": "altitude" }

description object 网关的自定义描述

antenna_placement object enum:indoor|outdoor

```
// response
{
    "code": "0",
    "data": ""
}
```

请求示例

```
curl --request PATCH \
    --url {host}/1.0/gateway/:gateway_eui \
    --user '<username>:<password>' \
    --header 'content-type: application/x-www-form-urlencoded' \
    --data '{"dev_name": "Example for Developer"}' \
    --include
```

从组织中删除网关

DELETE {host}/1.0/gateway/:gateway_eui

删除某个网关与组织账号的绑定关系，删除后，还可以通过SenseCAP 的APP将网关绑定回来。

Path Parameters

gateway_eui (required) string 设备的唯一标识

```
// response
{
    "code": "0",
    "data": ""
}
```

请求示例

```
curl --request DELETE \  
--url {host}/1.0/gateway/:gateway_eui \  
--user '<username>:<password>'
```

获取节点列表

GET {host}/1.0/lists/node

获取节点的列表

Query Parameters

offset number 当前页码，从1开始，默认为1

length string 每页获取的数量，默认为20

```
// response
{
    "code": "0",
    "data": {
        "page": {
            "count": "28" // total number
        },
        "list": [
            {
                "dev_eui": "2CF7F12004100005", // node EU
                "dev_name": "Example for Dev", // node name
                "dev_area": "Europe", // node area, Europe America Asia
                "dev_country": "CHN", // name country, defined by customer
                "online_status": "1", // 1 online, 0 offline
                "battery_status": "1", // 1 node battery good, 0 node battery bad
                "description": "describe", // desc something
                "activate_time": "1543990216000", // when the customer activates the device, unit milliseconds
                "latestmsg_time": "1543990216000", // the latest message time, unit milliseconds
                "production_time": "1539747273000", // the node production time, unit milliseconds
                "frequency": "EU_863_870", // frequency
                "msg_count": "12", // the total number of message that have been sent by the device
                "ver_hardware": "1.3", // hardware version
                "ver_software": "1.2" // software version
            }
        ]
    }
}
```

请求示例

```
curl --request GET \
--url {host}/1.0/lists/node?offset=1&length=2 \
--user '<username>:<password>'
```

获取节点详情

GET {host}/1.0/node/:node_eui

获取指定节点的详情

Path Parameters

node_eui (required) string 设备唯一标识

```

// response
{
    "code": "0",
    "data": {
        "dev_eui": "2CF7F12004100005",           // node EUI, equipment unique identity
        "dev_name": "Example for Dev",          // node name
        "dev_area": "Europe",                  // node area, Europe America Asia
        "dev_country": "CHN",                  // name country, defined by customer
        "group_name": "Group for Dev",         // the group name, if customer set a group for the device
        "access_read": "9872A8956A444FEF6A2B08625534D311F444714
FE5887AF338E42456EEDD8E16",      // device access code for read
        "access_write": "3D4474E08EDC039EB3EC2C3827666446D8D6C2
D31741F8FA271AD9AFA6B4A12D",     // device access code for write
        "online_status": "1",                  // 1 online, 0 offline
        "battery_status": "1",                // 1 node battery fine or good, 0 node battery for bad
        "description": "describe",           // desc something
        "activate_time": "1543990216000",     // when the customer activates the device, unit milliseconds
        "latestmsg_time": "1543990216000",   // the latest message time, unit milliseconds
        "production_time": "1539747273000",   // the node production time, unit milliseconds
        "frequency": "EU_863_870",            // frequency
        "msg_count": "12",                  // the total number of message that have been sent by the device
        "ver_hardware": "1.3",               // hardware version
        "ver_software": "1.2",              // software version
        "sensors": [                         // the sensors mounted on the device, some devices can mounted more than one sensor
            {
                "sensor_eui": "2CF7F13004200016",           // sensor EUI
                "sensor_name": "Temperature and Humidity", // sensor name
                "sensor_channel": "1",                      // the channel name or channel tag on which the sensor is mounted on the device
                "sensor_measure": [                         // some sensor have more than one measure

```

```
{  
    "id": "4097", //  
    "measure_id": "Air Temperature", //  
    "measure_name": "Air Humidity"  
}
```

请求示例

```
curl --request GET \  
      --url {host}/1.0/node/:node_eui \  
      --user '<username>:<password>'
```

更新节点详情信息

PATCH {host}/1.0/node/:node_eui

更新节点详情信息，以下Body Paramerters需选填任意一个或多个

Path Parameters

node_eui (required) string 设备唯一标识

Body Parameters

dev_name string 设备名称

group_unique_name string

想移动到的组的group_unique_name

description	string	节点的自定义描述
--------------------	--------	----------

```
// response
{
    "code": "0",
    "data": ""
}
```

请求示例

```
curl --request PATCH \
    --url {host}/1.0/gateway/:gateway_eui \
    --user '<username>:<password>' \
    --header 'content-type: application/x-www-form-urlencoded' \
    --data '{"dev_name": "Example for Developer"}' \
    --include
```

从组织中删除节点

DELETE {host}/1.0/node/:node_eui

删除某个节点与组织账号的绑定关系，删除后，可以通过SenseCAP 的APP将节点进行重新绑定。

Path Parameters

node_eui (required)	string	设备唯一标识
----------------------------	--------	--------

```
// response
{
    "code": "0",
    "data": ""
}
```

请求示例

```
curl --request DELETE \  
--url {host}/1.0/node/:node_eui \  
--user '<username>:<password>' \  

```

获取传感器测量值列表

GET {host}/1.0/lists/sensor/measure

获取所有传感器的物理测量值列表，同时你也可以在附录 - 传感器类型表中查看到这个信息。

```
// response
{
  "code": "0",
  "data": [
    {
      "sensor_id": "1009",
      "sensor_name": "Wind Speed",
      "sensor_measure": [
        {
          "measure_id": "4112",
          "measure_name": "Wind Direction",
          "measure_unit": ""
        },
        {
          "measure_id": "4105",
          "measure_name": "Wind Speed",
          "measure_unit": "m/s"
        }
      ]
    },
    {
      "sensor_id": "1008",
      "sensor_name": "Wind Direction",
      "sensor_measure": [
        {
          "measure_id": "4104",
          "measure_name": "Wind Direction",
          "measure_unit": ""
        }
      ]
    },
    {
      "sensor_id": "1006",
      "sensor_name": "Soil temperature and humidity",
      "sensor_measure": [
        {
          "measure_id": "4103",
          "measure_name": "Soil Humidity",
          "measure_unit": "%RH"
        },
        {
          "measure_id": "4102",
          "measure_name": "Soil Temperature",
          "measure_unit": "°C"
        }
      ]
    }
  ]
}
```

```
        ],
    },
    {
        "sensor_id": "1005",
        "sensor_name": "Atmospheric Pressure",
        "sensor_measure": [
            {
                "measure_id": "4101",
                "measure_name": "Atmospheric Pressure",
                "measure_unit": "Pa"
            }
        ]
    },
    {
        "sensor_id": "1004",
        "sensor_name": "Carbon Dioxide",
        "sensor_measure": [
            {
                "measure_id": "4100",
                "measure_name": "Carbon Dioxide",
                "measure_unit": "Vol"
            }
        ]
    },
    {
        "sensor_id": "1003",
        "sensor_name": "Light Intensity",
        "sensor_measure": [
            {
                "measure_id": "4099",
                "measure_name": "Light Intensity",
                "measure_unit": "Lux"
            }
        ]
    },
    {
        "sensor_id": "1001",
        "sensor_name": "Air Temperature and Humidity",
        "sensor_measure": [
            {
                "measure_id": "4098",
                "measure_name": "Air Humidity",
                "measure_unit": "%RH"
            },
            {

```

```
        "measure_id": "4097",
        "measure_name": "Air Temperature",
        "measure_unit": "°C"
    }
]
}
],
"time": 1.958
}
```

请求示例

```
curl --request GET \
--url {host}/1.0/lists/sensor/measure \
--user '<username>:<password>' \
```

设备数据API

设备数据API

GET {host}/1.0/lists/devices/eui

这是一个获取组织账号下所有设备EUI的便捷方法。

返回示例

```
{  
    "code": "0",  
    "data": {  
        "gateway": [  
            "2CF7F1.....",  
            "2CF7F1.....",  
            "2CF7F1....."  
        ],  
        "node": [  
            "2CF7F1.....",  
            "2CF7F1.....",  
            "2CF7F1.....",  
            "2CF7F1.....",  
            "2CF7F1....."  
        ]  
    }  
}
```

请求示例

```
curl --request GET \  
--url {host}/1.0/lists/devices/eui \  
--user '<username>:<password>'
```

获取设备最新数据

GET {host}/1.0/devices/data/:node_eui/latest

一个节点可以有多个通道，每个通道是用于连接传感器的物理接口。传感器可以产生多个测量值。例如温湿度传感器会输出温度值和湿度值。我们称这个温度值和湿度值为两个测量值。这个API可以从安装在某个指定channel的传感器中获取每个测量值的最新值。如果不传channel参数，就会返回这个设备的所有通道的所有传感器测量值。如果设备之前安装过其他传感器，也会在返回结果中包含以前的传感器测量值。

Path Parameters

node_eui (required) string 节点设备唯一标识

Query Parameters

measure_id string 传感器的测量值ID

channel string

返回该通道下的测量值，
如果不传参则默认返回所有通道的测量值

返回示例

```
{  
    "code": "0",  
    "data": [  
        {  
            "channel": "1",  
            "points": [  
                {  
                    "name": "Soil Temperature",  
                    "unit": "°C",  
                    "value": "13.75",  
                    "created": "1545013573"  
                },  
                {  
                    "name": "Soil Humidity",  
                    "unit": "%RH",  
                    "value": "19",  
                    "created": "1545013573"  
                }  
            ]  
        }  
    ]  
}
```

请求示例

```
curl --request GET \  
      --url {host}/1.0/devices/data/:node_eui/latest?measure_i  
d:measure_id&channel:channel \  
      --user '<username>:<password>' \  
      --include
```

获取设备历史数据

GET {host}/1.0/devices/data/:node_eui/raw

获取指定传感器节点设备的历史数据。

返回条数最多为400条。如果指定时间范围内的数据大于400条，则返回该时间范围内最新的400条数据。如果不传时间范围参数，将直接返回最新数据。

Path Parameters

node_eui (required)	string	节点设备唯一标识
----------------------------	--------	----------

Query Parameters

measure_id	string	传感器的测量值ID
-------------------	--------	-----------

channel	string
----------------	--------

返回该通道下的测量值，
如果不传参则默认返回所有通道的测量值

limit	number	要查询的记录条数，最多400条
--------------	--------	-----------------

time_start	number	时间戳，单位是毫秒
-------------------	--------	-----------

time_end	number	时间戳，单位是毫秒
-----------------	--------	-----------

```
// response
/*
数据集由通道和测量值ID标识，这意味着如果相同的通道安装过具有不同测量值ID的不同传感器，则可能返回多个数据集。例如通道1安装了光照传感器，并且在此之前安装过二氧化碳传感器，将有可能导致有两个与通道1相关的数据集 - {"channel": "1", "name": "Light", "points": [...], ...} and {"channel": "1", "name": "CO2", "points": [...], ...}。
*/
{
  "code": "0",
  "data": [
    {
      "channel": "1", // channel number
      "channel_name": "channel alias name", // measurement name
      "name": "Air Temperature", // measurement unit
      "unit": "°C", // measurement data points
      "points": [
        {
          "value": "8", // measurement value
          "created": "1545990300348" // timestamp, unit millisecond
        },
        {
          "value": "7",
          "created": "1545989996049"
        }
      ]
    },
    {
      "channel": "2",
      "channel_name": "the 2nd channel",
      "name": "Air Humidity",
      "unit": "%RH",
      "points": [
        {
          "value": "59",
          "created": "1545990300348"
        },
        {
          "value": "68",
          "created": "1545989996049"
        }
      ]
    }
  ]
}
```

```
        "value": "68",
        "created": "1545989996060"
    }
]
}
],
"time": 0.297
}
```

请求示例

```
curl --request GET \
--url {host}/1.0/devices/data/:node_eui/raw?limit=:limit&tim
e_start=:time_start&time_end=:time_end \
--user '<username>:<password>' \
--include
```

获取设备数据段

GET {host}/1.0/devices/data/:node_eui/segment

将庞大的数据段分成小数据段，然后输出每个小段的平均值和峰值。用时间长度描述小段，例如：60分钟。

Path Parameters

node_eui (required) string 节点设备唯一标识

Query Parameters

measure_id string 传感器的测量值ID

channel string

返回该通道下的测量值，
如果不传参则默认返回所有通道的测量值

limit string 要查询的记录条数，最多400条

time_start string 时间戳，单位是毫秒

time_end string 时间戳，单位是毫秒

返回示例

```
{  
    "code": "0",  
    "data": [  
        {  
            "channel": "1", // channale number  
            "channel_name": "channel alias name",  
            "name": "Wind Direction", // measurement name  
            "unit": "", // measurement unit  
            "list": [  
                {  
                    "created": "1545181200", // timestamp, unit second  
                    "avg": "30", // avgrage value of the segment  
                    "peak": "90" // peak value of the segment  
                },  
                {  
                    "created": "1545182200",  
                    "avg": "31",  
                    "peak": "93"  
                }  
            ]  
        }  
    ]  
}
```

请求示例

```
curl --request GET \
--url {host}/1.0/devices/data/:node_eui/segment?measure_id:me
asure_id&channel:channel&segment:segment&time_start:time_star
t&time_end:time_end \
--user '<username>:<password>' \
--include
```

Data OpenStream API快速入门

Summary:

本指南将向您展示如何订阅设备的消息，以及如何使用Eclipse Mosquitto's CLIS订阅或发布消息，将命令发送到特定的设备中。

准备工作

- 下载 并安装Mosquitto . .

获取证书

1. 登录SenseCAP云平台：<https://sensecap.seeed.cc>
2. 前往“Organization/Security Credentials”
3. 点击“Create access key”会产生一对Access ID和对应的Access Key

The screenshot shows the SenseCAP cloud platform's 'Access API keys' section. It includes a note about using access keys for secure requests, the organization ID (521853156991), and a 'Create access key' button. Below this, a table lists an access key entry:

Creation Time	Label	Access Key ID	Access Level	Status
2019-06-24 15:59:45		52015BC0B062617B	<input type="radio"/> Only Read <input checked="" type="radio"/> Full Access	Active (effect at 2019-06-24 15:59:45)

4. 点击“Organization”，获取“Organization ID”作为。

The screenshot shows the 'Organization' tab selected in the Sensecap OpenAPI interface. It displays the following information:

- Owner information:**
 - Seed Account: iot@seeed.cc
 - Owner: Seed
 - Creation Time: 2018-12-29 06:49:04
- Location:**
 - Country: CHN
 - State: Shenzhen
 - Address: 9F, Building G3, TCL International E City, ZhongShanYuan Road, NanShan, Shenzhen, GuangDong Province, China
- Business:**
 - Organization Name: Seed
 - Email: iot@seeed.cc
 - Telephone: +86-0755-36852963
 - Official Website: <https://www.seeed.cc>
- Organization Id:** 521 (highlighted with a red border)

接收设备消息

下面介绍如何接收您所有设备的消息 1.打开一个终端窗口并执行以下命令。

```
mosquitto_sub \
-h openstream.api.sensecap.seeed.cc \
-t '/device_sensor_data/<OrgID>/+/+/+/+' \
-u 'org-<OrgID>' \
-P '<Password>' \
-I 'org-<OrgID>-quickstart' \
-v
```

请将您刚获取到的Organization ID和Access Key将上面的和替换掉。

2.启动设备，当设备持续发送消息时，信息的输出格式如下：

```
/device_sensor_data/<OrgID>/<DeviceEUI>/<Channel>/<SensorEU
I>/<MeasurementID>
```

```

/device_sensor_data/xxxx/2CF7F12XXXXXXXXX/1/2CF7F13XXXXXXXXX/41
05 {"value":0,"timestamp":1544151824139}
/device_sensor_data/xxxx/2CF7F12XXXXXXXXX/1/2CF7F13XXXXXXXXX/40
97 {"value":23,"timestamp":1544151900992}
/device_sensor_data/xxxx/2CF7F12XXXXXXXXX/1/2CF7F13XXXXXXXXX/41
01 {"value":101629,"timestamp":1544151901112}
/device_sensor_data/xxxx/2CF7F12XXXXXXXXX/1/2CF7F13XXXXXXXXX/40
98 {"value":71,"timestamp":1544151900992}
/device_sensor_data/xxxx/2CF7F12XXXXXXXXX/1/2CF7F13XXXXXXXXX/40
99 {"value":69.12,"timestamp":1544151902224}
/device_sensor_data/xxxx/2CF7F12XXXXXXXXX/1/2CF7F13XXXXXXXXX/41
00 {"value":437,"timestamp":1544151922137}

```

显示传感器收集的每个数据，包括设备唯一标识Device EUI，设备通道Device Channel，传感器唯一标识Sensor EUI，测量值IDMeasurement ID，测量值Measurement Value和时间戳timestamp。

订阅特定字段

将设备通道Device EUI，设备通道Device Channel，传感器唯一标识Sensor EUI，测量值ID Measurement Value替换成从上述消息列表中的特定设备的字段，以便从某个设备或某个通道获取数据。

比如订阅温湿度传感器设备ID为2CF7F12210400083，通道为1，传感器ID为2CF7F13004700089，温度物理量ID为4097的值，用Organization ID和Access Key将下面的和替换掉，在shell中执行命令

```

mosquitto_sub \
-h openstream.api.sensecap.seeed.cc \
-t '/device_sensor_data/<OrgID>/2CF7F12XXXXXXXXX/#' \
-u '<org-<OrgID>>' \
-P '<Password>' \
-I '<org-<OrgID>-quickstart>' \
-v

```

执行命令的结果如下：

```

/device_sensor_data/521853156991/2CF7F12210400083/1/2CF7F130047
0089/4097 {"value":28,"timestamp":1561373812474}

```

Data OpenStream API方法说明

连接信息

- Host: sensecap-openstream.seeed.cc
- 端口：MQTT的端口：1883；MQTT Over WebSocket的端口：8083。
- Client ID：格式为org-<Organization ID>-<Random ID>, <Orgnization ID>可以在您的后台Organization页面中获取, <Random ID> 使用您自己随机生成的数字和小写字母。
- Username：格式为org-<Organization ID>，<Organization ID>可以在您的后台Organization中获取。
- Password：在Web页面获取Access Key

发布和订阅模型

SenseCAP Data OepnStream API 服务实现了标准化的发布/订阅模型，平台通过 MQTT代理实现了该模型，并具有端到端的消息交互通信功能。该服务还支持MQTT Over Websocket标准协议。

通过publish方法，您可以向设备发送控制命令。通过subscribe方法，您可以订阅并监听设备上报的传感器遥测数据消息。

消息主题

接收所有设备的遥测数据

/device_sensor_data/<OrgID>/+/+/+/+

接收指定设备的遥测数据

主题格式：

/device_sensor_data/<OrgID>/<DeviceEUI>/<Channel>/<SensorEUI>/<MeasurementID>

字段	描述
OrgID	您的组织ID，可以从后台中的Organization页面中获取。
DeviceEUI	传感器设备唯一标识

字段	描述
Channel	一个设备上的物理接口，用于连接传感器，默认为1
SensorEUI	传感器探头的唯一标识
MeasurementID	测量值ID，具体测量值的清单详见附录

Note: “+”表示此字段没有过滤条件，能够匹配所有。所以，“/+ / + / + / +”表示监听所有“<DeviceEUI>”，“<Channel>”，“<SensorEUI>”，“<MeasurementID>”

主题可以指定过滤条件，以实现对指定设备，通道和远程测量数据类型的监听。例如，如果您只想侦听设备ID为“2F000000000000”的设备，可以用2F000000000000替换字段<DeviceEUI>。

上面的“2F000000000000”设备必须是已经绑定到您组织账号中的设备。而且<OrgID>字段是必填的。

消息主体

新的测量数据

```
{
  "value": 437,
  "timestamp": 1544151922137
}
```

这是一个由设备上传的传感器远程测量数据消息，其符合JSON格式并且可以被JSON解析器解析。通常，对于大多数功能需求，主体需要与主题中的某些字段结合一起使用。

字段	描述
value	传感器的测量数值
timestamp	数据的时间戳，单位为毫秒

传感器的类型表

传感器ID	传感器名称	测量值ID (Measurement ID)	测量值名称
1011	SenseCap SenseCAP雨量传感器	0x1011(4113)	Rainfall(hour)
100E	SenseCAP SenseCAP土壤温度、VWC、EC传感器	0x1010(4112), 0x100F(4111), 0x100E(4110)	Soil Temperature, Soil Electrical Conductivity, Soil Volumetric Water Content
100D	SenseCAP SenseCAP溶解氧传感器	0x100D(4109)	Dissolved Oxygen
100C	SenseCAP SenseCAP EC传感器	0x100C(4108)	Electrical Conductivity
100B	SenseCAP SenseCAP 光通量传感器	0x100B(4107)	Light quantum
100A	SenseCAP SenseCAP pH值传感器	0x100A(4106)	Water PH
1009	SenseCAP SenseCAP 风速传感器	0x1009(4105)	Wind Speed
1008	SenseCAP SenseCAP 风向传感器	0x1008(4104)	Wind Direction
1006	SenseCAP SenseCAP 土壤温湿度传感器	0x1007(4103), 0x1006(4102)	Soil Humidity, Soil Temperature
1005	SenseCAP SenseCAP 气压传感器	0x1005(4101)	Air Pressure
1004	SenseCAP SenseCAP 二氧化碳传感器	0x1004(4100)	CO2

传感器ID	传感器名称	测量值ID (Measurement ID)	测量值名称
1003	SenseCAP 光照传感器	0x1003(4099)	Light
1001	SenseCAP 空气温湿度传感器	0x1002(4098), 0x1001(4097)	Air Humidity, Air Temperature

测量值信息表

测量值ID	测量值名称	数据类型	数值范围	单位
0x1011(4113)	雨量(小时)	number	0~240	mm/hour
0x1010(4112)	土壤温度	number	-40~60	°C
0x100F(4111)	电导率	number	0~23	ds/m ²
0x100E(4110)	土壤体积含水量	number	0~100	%
0x100D(4109)	溶解氧	number	0~20	mg/L
0x100C(4108)	电导率	number	0~23	dS/m
0x100B(4107)	光通量	number	0~2000	umol/m ² s
0x100A(4106)	水PH值	number	0~14	PH
0x1009(4105)	风速	number	0~60	m/s
0x1008(4104)	风向	Enum	0,45,90,135,180,225,270,315	°
0x1007(4103)	土壤湿度	number	0~100	%RH
0x1006(4102)	土壤温度	number	-30~70	°C
0x1005(4101)	气压	number	300~1100000	Pa
0x1004(4100)	二氧化碳	number	400~10000	ppm
0x1003(4099)	光照	number	0~188000	Lux
0x1002(4098)	空气湿度	number	0~100	%RH
0x1001(4097)	空气温度	number	-40~90	°C