

Sensecap Open Api 文档

SenseCAP API 简介

简介

HTTP API

HTTP API 快速入门

HTTP API 访问指南

HTTP API 方法说明

组织管理API

组管理API

设备管理API

设备数据API

Data OpenStream API

Data OpenStream API快速入门

Data OpenStream API方法说明

SenseCAP SDK

Java的SDK

附录

传感器的类型表

测量值信息表

设备状态ID表

错误码列表

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。详细产品信息可以访问：<https://www.seeedstudio.com.cn>

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云平台提供配套API，用户可以从云平台上获取数据并做二次集成开发。

SenseCAP API是专门用于用户做设备管理和数据调用的接口，它由HTTP API和Data OpenStream API组成：

- 基于HTTP API，用户可以管理SenseCAP所有设备，获取原始数据或者历史数据。
- 基于Data OpenStream API，用户可以实时获取传感器的测量数据。

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

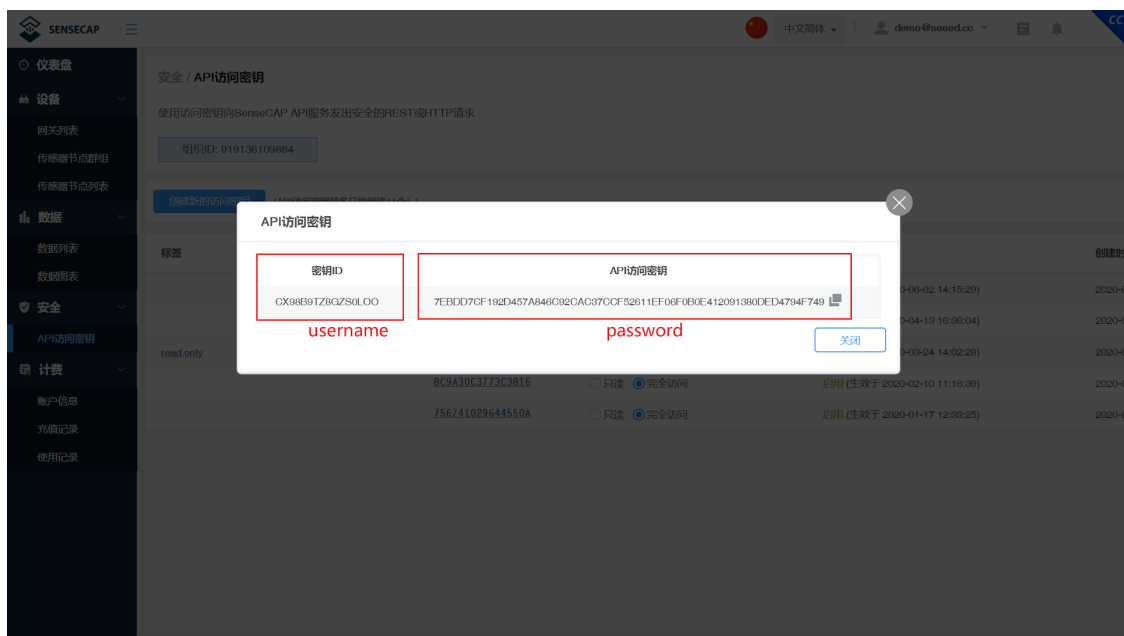
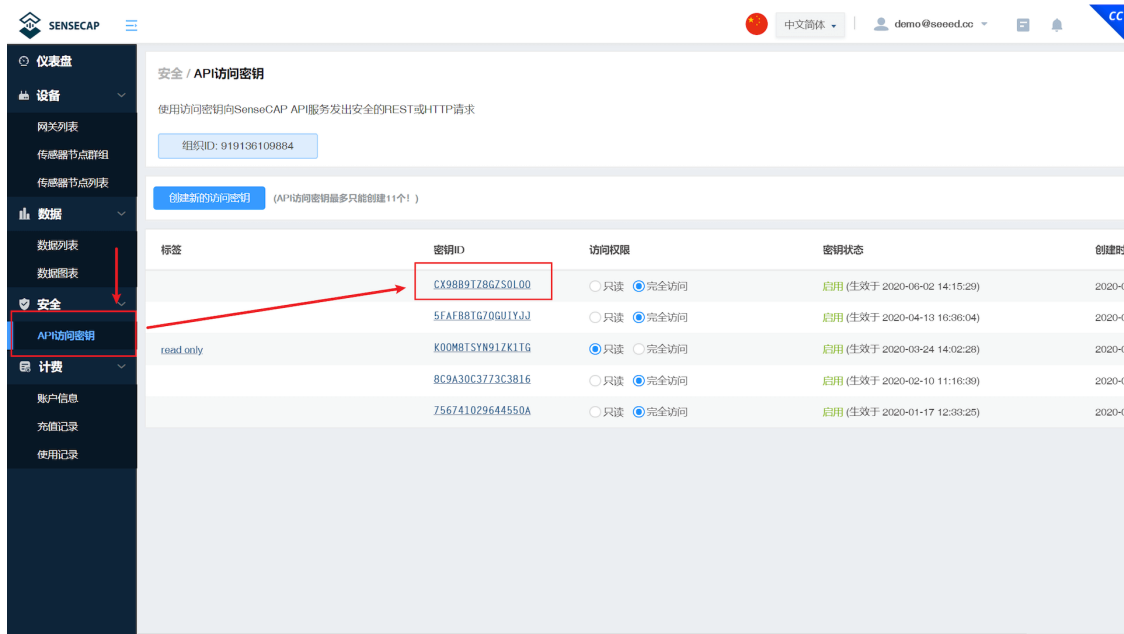
若您还没有账号，请先注册SenseCAP云平台账号，云平台地址：

- 中国站：<https://sensecap.seeed.cn>
- 国际站：<https://sensecap.seeed.cc>

提示：若您的设备类型是LoRaWAN，必须使用国际站。

获取Access Key

1. 登录SenseCAP云平台。
2. 点击左导航栏“安全/API访问密钥”。
3. 点击“创建新的访问密钥”。
4. 点击“密钥ID”，输入密码后获取“密钥ID”和“API访问密钥”。



获取账号下的分组信息

使用curl发出HTTP请求。下面的示例将调用API来获取账号下的所有设备分组。

- username = 密钥ID
- password = API访问密钥

```
curl --user "<username>:<password>" \
https://sensecap.seeed.cc/openapi/list_groups
```

请将您刚获取到的密钥ID和API访问密钥将示例的<username>和<password>替换掉。返回内容如下：

```
{
  "code": "0",
  "data": [
    {
      "group_name": "Default",
      "group_uuid": ""
    },
    {
      "group_name": "test group",
      "group_uuid": "80523B280630E611"
    },
    {
      "group_name": "demo",
      "group_uuid": "EBAD5387C4FC8711"
    }
  ]
}
```

HTTP API 访问指南

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 HOST

- 中国站: <https://sensecap.seeed.cn/openapi>
- 国际站: <https://sensecap.seeed.cc/openapi>

HTTP HEADER

请求

字段名称	说明
API-VERSION	接口版本号

响应

字段名称	说明
api-gateway-execute-second	执行该接口所花费的时间，单位秒
api-gateway-mpuoc-consume	执行该接口所消耗的配额（详细规则可查看简介中的API配额规则介绍）

HTTP基本身份验证

我们使用Access ID（密钥ID）作为用户名，Access Key（API 访问密钥）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云平台来获取密钥信息。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"  
}
```

Organization Management API

Get organization information

GET {host}/view_organization

Get organization information

Response

Name	Description
org_id	organization id

Example request

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

{
  "code": "0",
  "data": {
    "org_id": "919136109886",
  }
}
```

组管理API

创建设备组

POST {host}/create_group

此功能用于创建新设备组，上限为50组。

请求

Body Parameters

name (required) string 自定义组名称

响应

字段名称	说明
group_name	设备组名称
group_uuid	设备组唯一标识

请求示例

```
curl --request POST \
  --url '{host}/create_group' \
  --user '<username>:<password>' \
  --header 'Content-Type: application/json; charset=utf-8' \
  --data '{"name":"device group name"}' \
  --include

{
  "code": "0",
  "data": {
    "group_name": "test group",
    "group_uuid": "478ED9E2A290F401"
  }
}
```

接口历史版本

历史版本	说明
Sensecap V1	已停止维护，不推荐使用

更新设备组信息

POST {host}/update_group

更新设备组名称

请求

Body Parameters

group_uuid (required) string 设备组唯一标识

name (required) string 分组名称

请求示例

```
curl --request POST \
  --url '{host}/update_group' \
  --user '<username>:<password>' \
  --header 'Content-Type: application/json; charset=utf-8' \
  --data '{"group_uuid":"259B1906C3B38481", "name":"new name of the device group"}' \
  --include

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

删除设备组

POST {host}/delete_group

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

请求

Query Parameters

group_uuid (required) string 设备组唯一标识

请求示例

```
curl --request POST \
  --url '{host}/delete_group?group_uuid=0C32119F38C89C31' \
  --user '<username>:<password>' \
  --header 'content-type: application/json' \
  --include

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

获取设备组列表

GET {host}/list_groups

获取所有设备组

响应

字段名称	说明
group_name	设备组名称
group_uuid	设备组唯一标识

请求示例

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

{
  "code": "0",
  "data": [
    {
      "group_name": "Default",
      "group_uuid": ""
    },
    {
      "group_name": "test group",
      "group_uuid": "80523B280630E611"
    },
    {
      "group_name": "demo",
      "group_uuid": "EBAD5387C4FC8711"
    }
  ]
}
```

接口历史版本

历史版本	说明
Sensecap V1	已停止维护，不推荐使用

移动设备到其它组

POST {host}/move_devices_to_group
将设备移动到某个分组中

请求

Body Parameters

group_uuid (required)	string	设备组唯一标识
devices (required)	array	设备唯一标识

请求示例

```
curl --request POST \
  --url '{host}/move_devices_to_group' \
  --user '<username>:<password>' \
  --header 'Content-Type: application/json; charset=utf-8' \
  --data '{"devices":["2CF7F12004100001","2CF7F12004100002"],"group_uuid": "0BC724D9F32AD934"}' \
  --include

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

设备管理API

获取设备列表

GET {host}/list_devices

获取设备列表

请求

Query Parameters

device_type	string	设备类型: 1-gateway, 2-node(默认值)
group_uuid	string	设备组唯一标识,默认返回默认分组,不传则返回所有分组

响应

字段名称	说明
device_eui	设备唯一标识
device_name	设备名称

请求示例

```
curl --request GET \
  --url {host}/list_devices?device_type=1&group_uuid=0C32119F38C89C31 \
  --user 'username:password'
```

```
{
  "code": "0",
  "data": [
    {
      "device_eui": "2CF7F12010700088",
      "device_name": "设备2CF7F12010700088"
    },
    {
      "device_eui": "2CF7F1201070001C",
      "device_name": "设备2CF7F1201070001C"
    },
    {
      "device_eui": "2CF7F12104700010",
      "device_name": "US915-2CF7F12104700010"
    }
  ]
}
```

接口历史版本

历史版本	说明
Sensecap V1	已停止维护, 不推荐使用

获取设备详情

POST {host}/view_devices

获取设备详情信息

请求

Body Parameters

device_euis (required)	array	设备唯一标识,一次最多查询50个设备
device_type	string	设备类型: 1-gateway, 2-node(默认值)

响应

字段名称	说明
frequency	设备通信频率
device_eui	设备唯一标识
device_name	设备名称
device_network	组网协议, 1:LoRaWAN, 2: NB-IoT, 3: 2G , 4:LoRaPP
position	设备GPS位置, latitude: 经度, latitude: 纬度
position_source	GPS位置来源, 0-手动设置的位置, 1-设备上报的位置
hardware_version	设备硬件版本号
software_version	设备软件版本号
sim	设备上的sim卡信息
iccid	ICCID
msisdn	MSISDN
activateTime	激活日期
expiryDate	计费结束日期
status	状态, 0-未知, 1-正常, 2-单项停机, 3-停机, 4-预销号, 5-销号, 6-过户, 7-休眠, 8-待激活
flow	当月已用流量
residueFlow	剩余流量

请求示例

```
curl --request POST \
  --url {host}/view_devices \
  --data '{"device_type":1, "device_euis":["2CF7F15000100122"]}' \
  --header 'Content-Type: application/json; charset=utf-8' \
  --user '<username>:<password>'

{
  "code": "0",
  "data": [
    {
      "frequency": "470",
      "device_eui": "2CF7F15000100122",
      "device_name": "设备2CF7F15000100122",
      "device_network": 2,
      "position": {
        "latitude": 113.931225,
        "longitude": 22.569792
      },
      "position_source": 0,
      "hardware_version": "",
      "software_version": "23.0",
      "sim": {
        "iccid": "89860446091891237424",
        "msisdn": "1440467057424",
        "activateTime": "2019-12-03T00:00:00.000Z",
        "expiryDate": "2020-11-30T00:00:00.000Z",
        "status": 1,
        "flow": 0,
        "residueFlow": 1024
      }
    }
  ]
}
```

接口历史版本

历史版本	说明
Sensecap V1	已停止维护, 不推荐使用

获取节点通道列表

POST {host}/list_device_channels
获取设备通道列表

请求

Body Parameters

device_euis (required) array 设备唯一标识,一次最多查询50个设备

响应

字段名称	说明
device_eui	设备唯一标识
channel_index	通道编号
sensor_id	传感器id
sensor_status	传感器状态:0-闲置 1-正常 2-异常

字段名称	说明
channel_type	通道类型:1: 485 Sensor; 2: Sseed Sensor; 3: 485 Output; 4: Sseed Output
sensor_type	传感器类型
channel_name	通道名称
measurement_ids	测量值id

请求示例

```
curl --request POST \
  --url {host}/list_device_channels \
  --data '{"device_euis":["2CF7F15000100147","2CF7F16221200060"]}' \
  --header 'Content-Type: application/json; charset=utf-8' \
  --user '<username>:<password>'

{
  "code": "0",
  "data": [
    {
      "device_eui": "2CF7F15000100147",
      "channels": [
        {
          "channel_index": 1,
          "sensor_id": "2CF7F13011900006",
          "sensor_status": 1,
          "channel_type": 2,
          "sensor_type": "1005",
          "channel_name": "",
          "measurement_ids": [
            "4101"
          ]
        }
      ]
    },
    {
      "device_eui": "2CF7F16221200060",
      "channels": [
        {
          "channel_index": 11,
          "sensor_id": "0111006221200060",
          "sensor_status": 1,
          "channel_type": 1,
          "sensor_type": "2001",
          "channel_name": "",
          "measurement_ids": [
            "4097",
            "4105"
          ]
        }
      ]
    }
  ]
}
```

获取设备工作状态

POST {host}/view_device_running_status
 获取设备工作状态

请求

Body Parameters

device_euis (required) array 设备唯一标识,一次最多查询50个设备

响应

字段名称	说明
device_eui	设备唯一标识
latest_message_time	设备最后上报消息的时间
online_status	在线状态:0-离线, 1-在线
battery_status	电池电量状态:0-低电量 1-电量良好
report_frequency	设备上报频率, 单位分钟, 返回-1则设备未上报该信息

请求示例

```
curl --request POST \
  --url {host}/view_device_running_status \
  --data '{"device_euis":["2CF7F1101300001C","2CF7F16221200060"]}' \
  --header 'Content-Type: application/json; charset=utf-8' \
  --user '<username>:<password>'

{
  "code": "0",
  "data": [
    {
      "device_eui": "2CF7F16221200060",
      "latest_message_time": "2020-04-20T07:06:32.944Z",
      "online_status": 0,
      "battery_status": 1,
      "report_frequency": 0
    },
    {
      "device_eui": "2CF7F1101300001C",
      "latest_message_time": "",
      "online_status": 0,
      "battery_status": 1,
      "report_frequency": -1
    }
  ]
}
```

获取传感器测量值列表

GET <https://sensecap-statics.seeed.cn/refer/def/sensor.json>

获取所有传感器的物理测量值列表, 传感器测量值参考列表以文件的形式存储于云端, 你可直接获取该文件。

字段说明

字段名称	说明
zh-cn	中文描述
en	英文描述
sm	记录了每个传感器类型对应的测量类型id

字段名称	说明
rg	测量值取值范围
sensorType	传感器类型，记录每个传感器类型的名称
measurementId	测量值id,记录每个测量值的名称及单位

示例

```
{
  "zh-cn": {
    "sensorType": {
      "1001": "空气温湿度传感器",
      "1003": "光照强度传感器",
      ...
    },
    "measurementId": {
      "4097": ["空气温度", "℃"],
      "4098": ["空气湿度", "%RH"],
      ...
    }
  },
  "en": {
    "sensorType": {
      "1001": "Air Temperature and Humidity Sensor",
      "1003": "Light Intensity Sensor",
      ...
    },
    "measurementId": {
      "4097": ["Air Temperature", "℃"],
      "4098": ["Air Humidity", "%RH"],
      ...
    }
  },
  "sm": {
    "1001": ["4098", "4097"],
    "1003": ["4099"],
    ...
  },
  "rg": {
    "4097": "-40~90",
    "4098": "0~100",
    ...
  }
}
```

接口历史版本

历史版本	说明
Sensecap V1	已停止维护，不推荐使用

绑定设备

POST {host}/bind_device

将设备绑定到账户下

请求

Body Parameters

eui (required)string设备eui

code (required)	string	设备code
device_name	string	设备名称
group_uuid	string	设备组唯一标识, 可通过群组列表接口获取
longitude	string	设备位置, 经度
latitude	string	设备位置, 纬度

请求示例

```
curl --request POST \
  --url '{host}/bind_device' \
  --user '<username>:<password>' \
  --header 'Content-Type: application/json; charset=utf-8' \
  --data '{"code":"device code","eui":"device eui"}' \
  --include

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

接口历史版本

历史版本	说明
Sensecap V1	已停止维护, 不推荐使用

设备解绑

POST {host}/delete_devices
解绑设备

请求

Body Parameters

device_euis (required)	array	设备唯一标识,一次最多解绑50个设备
-------------------------------	-------	--------------------

请求示例

```
curl --request POST \
  --url {host}/delete_devices \
  --data '{"device_euis":["2CF7F15000100122"]}' \
  --header 'Content-Type: application/json; charset=utf-8' \
  --user '<username>:<password>'

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

设备数据API

获取最新遥测数据

GET {host}/view_latest_telemetry_data

返回该设备一年内最新的遥测数据,如果没指定channel_index则返回设备每个通道下最新的数据点。

请求

Query Parameters

device_eui (required)	string	设备唯一标识
channel_index	string	返回该通道下的测量值
measurement_id	string	传感器的测量值ID

响应

字段名称	说明
channel_index	通道编号
measurement_value	测量值
measurement_id	测量值id
time	测量时间

请求示例

```
curl --request GET \
  --url {host}/view_latest_telemetry_data?device_eui=2CF7F12010700001&measurement_id=4104&channel_index=32 \
  --user 'username:password' \
  --include

{
  "code": "0",
  "data": [
    {
      "channel_index": 32,
      "points": [
        {
          "measurement_value": 185,
          "measurement_id": "4104",
          "time": "2020-03-26T07:28:28.575Z"
        }
      ]
    }
  ]
}
```

接口历史版本

历史版本	说明
Sensecap V1	已停止维护, 不推荐使用

获取历史遥测数据

GET

{host}/list_telemetry_data

获取指定设备的历史遥测数据。

获取指定传感器节点设备的历史数据，最长返回一个月的数据 只能查询最近三个月内的数据

请求

Query Parameters

device_eui (required)	string	设备唯一标识
channel_index	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
measurement_id	string	传感器的测量值ID
limit	number	要查询的记录条数
time_start	number	时间戳，单位是毫秒，默认一天前的时间
time_end	number	时间戳，单位是毫秒，默认当前时间

响应

返回格式采样列数据的表达方式

字段名称	说明
list[0]	通道编号及测量值id
list[1]	遥测数据，其中包含测量值及测量时间

请求示例

```
curl --request GET \
  --url {host}/list_telemetry_data?device_eui=2CF7F12210400082 \
  --user '<username>:<password>' \
  --include

{
  "code": "0",
  "data": {
    "list": [
      [[1, "4097"], [1, "4098"], [2, "4097"]],
      [
        [22, "2020-03-06T23:48:00Z"], [23, "2020-03-06T23:48:00Z"], [24, "2020-03-06T23:48:00Z"]],
        [
          [22, "2020-03-06T23:48:00Z"], [23, "2020-03-06T23:48:00Z"] ],
          [22, "2020-03-06T23:48:00Z"], [23, "2020-03-06T23:48:00Z"] ]
        ]
      ]
    }
  }
```

返回说明如下：

- list[0][0]：表示通道及测量值id,该例中通道为1,测量值id为4097
- list[1][0]：对应list[0][0]的测量结果
- list[1][1]：对应list[0][1]的测量结果

- list[1][0][0]: 该例中表示通道1对应的测量值id为4097的第一个测量结果, 测量值为22, 测量时间为2020-03-06T23:48:00Z

接口历史版本

历史版本	说明
Sensecap V1	已停止维护, 不推荐使用

获取设备遥测数据折线图

GET {host}/aggregate_chart_points

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, 最长返回一年的数据, 每个测量量最多返回250个点, 超过250个点将自动重新划分时间段返回250个点

请求

Query Parameters

device_eui (required)	string	设备唯一标识
channel_index	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
measurement_id	string	传感器的测量值ID
interval	string	要获取的时间段长度, 以分钟为单位,默认60分钟
time_start	string	时间戳, 单位是毫秒, 默认一天前的时间
time_end	string	时间戳, 单位是毫秒, 默认当前时间

响应

返回格式采样列数据的表达方式

字段名称	说明
channel	通道编号
average_value	测量值平均值
measurement_id	测量值id
time	测量时间

请求示例

```
curl --request GET \
  --url {host}/aggregate_chart_points?device_eui=2CF7F12210400082 \
  --user '<username>:<password>' \
  --include

{
  "code": "0",
  "data": [
    {
      "channel": 1,
      "lists": [
        {
          "average_value": 7.27,
          "measurement_id": "4106",
          "time": "2019-05-25T23:42:00.000Z"
        },
        {
          "average_value": 6.85,
          "measurement_id": "4106",
          "time": "2019-05-27T10:45:00.000Z"
        }
      ]
    }
  ]
}
```

接口历史版本

历史版本	说明
Sensecap V1	已停止维护，不推荐使用

Data OpenStream API快速入门

Summary:

本指南将向您展示如何订阅设备的信息，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：

- 中国站：<https://sensecap.sened.cn>
- 国际站：<https://sensecap.sened.cc>

Note: LoRaWAN设备必须使用国际站

2.前往“安全/API访问密钥”。

3.点击“Create access key”会产生一条密钥ID，点击密钥ID并输入账号密码后，可获取对应的API访问密钥（Password）。

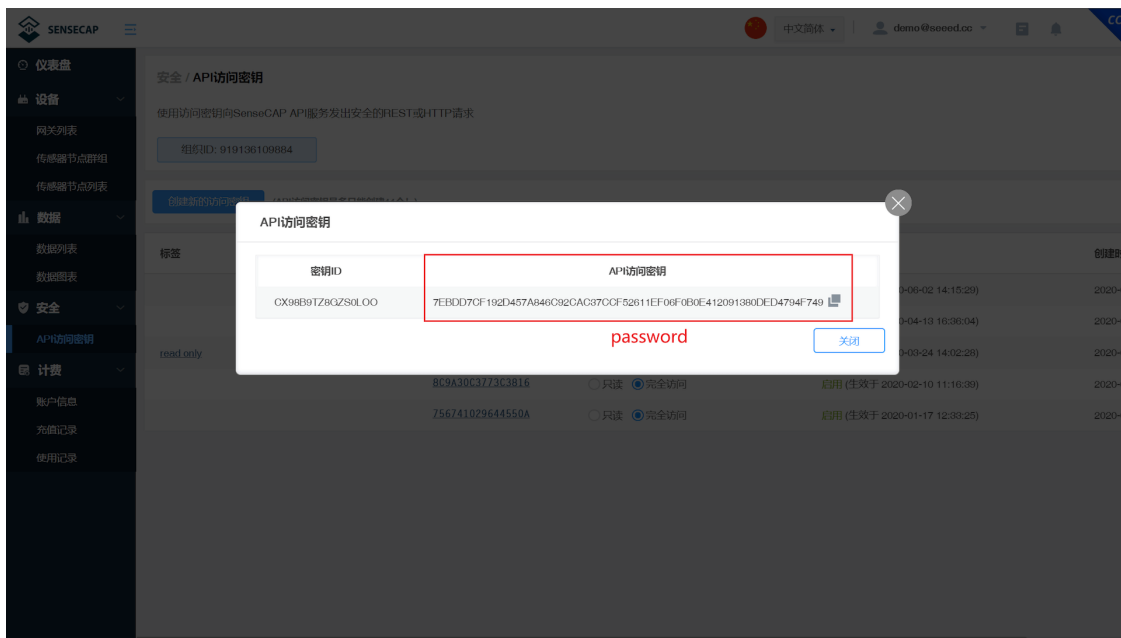
安全 / API访问密钥

使用访问密钥向SenseCAP API服务发出安全的REST或HTTP请求

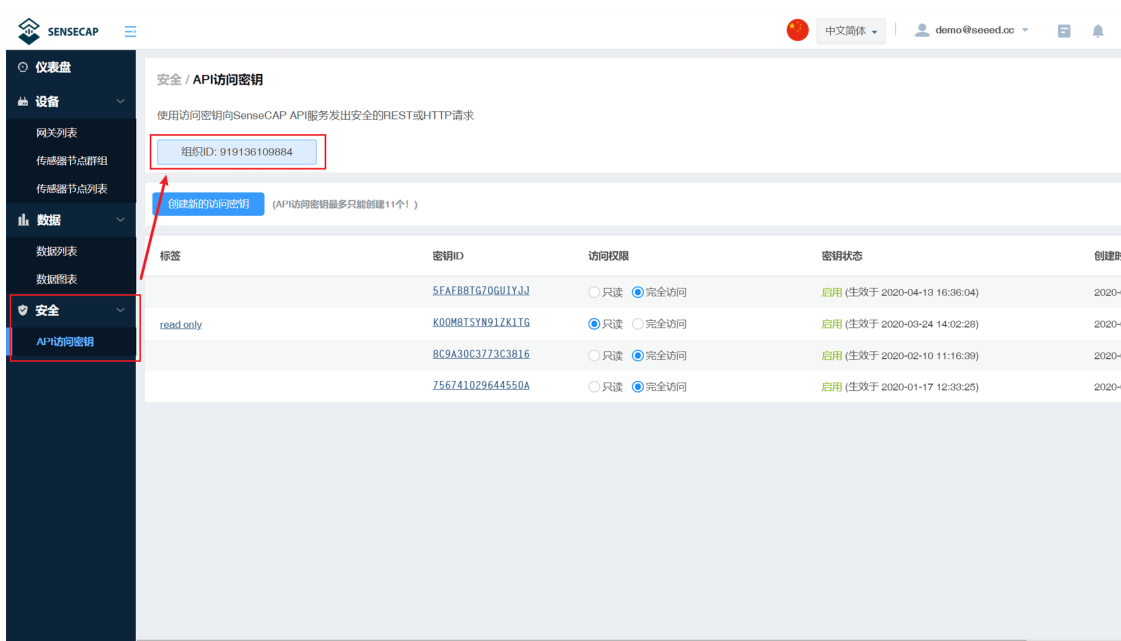
组织ID: 919136109884

[创建新的访问密钥](#) (API访问密钥最多只能创建11个!)

标签	密钥ID	访问权限	密钥状态	创建时间
	CX98891Z86ZS0L00	☐ 只读 ☒ 完全访问	启用 (生效于 2020-06-02 14:15:29)	2020-0
	5FAFB8IGZ0GUYJJ	☐ 只读 ☒ 完全访问	启用 (生效于 2020-04-13 16:36:04)	2020-0
read only	K00M81SYN91ZK1TG	☒ 只读 ☐ 完全访问	启用 (生效于 2020-03-24 14:02:28)	2020-0
	8C9A30C3773C3816	☐ 只读 ☒ 完全访问	启用 (生效于 2020-02-10 11:16:39)	2020-0
	Z56741029644550A	☐ 只读 ☒ 完全访问	启用 (生效于 2020-01-17 12:33:25)	2020-0



4. 点击“Organization”，获取“Organization ID”作为<OrgID>。



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。

- OrgID = 组织ID
- Password = API访问密钥

```
mosquitto_sub \
  -h sensecap-openstream.seeed.cn \
  -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(组织ID)和Access Key (API访问密钥) 将上面<OrgID>的和<Password>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>/<Reserved>/<MeasurementID>
```

```
/device_sensor_data/1234/2CF7F12000000001/1/vs/4105 {"value":2,"timestamp":1544151824139}
/device_sensor_data/xxxx/2CF7F12XXXXXXX/1/vs/4097 {"value":23,"timestamp":1544151900992}
/device_sensor_data/xxxx/2CF7F12XXXXXXX/1/vs/4101 {"value":101629,"timestamp":1544151901112}
/device_sensor_data/xxxx/2CF7F12XXXXXXX/1/vs/4098 {"value":71,"timestamp":1544151900992}
/device_sensor_data/xxxx/2CF7F12XXXXXXX/1/vs/4099 {"value":69.12,"timestamp":1544151902224}
/device_sensor_data/xxxx/2CF7F12XXXXXXX/1/vs/4100 {"value":437,"timestamp":1544151922137}
```

以第一包数据为例	字段	释义
1234	OrgId	账号的组织ID
2CF7F12000000001	DeviceEUI	设备唯一标识
1	Channel	设备通道
vs	Reserved	预留字段
4105	MeasureID	测量值类型，此处4105是风速
2	value	采集的测量值，即风速为2m/s
1544151824139	timestamp	数据的采集时间戳

订阅特定字段

指定特定字段可实现订阅某个具体设备或通道的数据。

举例：订阅温湿度传感器采集的温度值。

- DeviceEUI = 2CF7F12210400083
- Channel = 1
- MeasureID = 4097
- OrgID = 组织ID
- Password = API访问密钥

替换相关参数后，执行命令：

```
mosquitto_sub \
  -h sensecap-openstream.seeed.cn \
  -t '/device_sensor_data/<OrgID>/2CF7F12210400083/1/vs/4097' \
  -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/vs/4097 {"value":28,"timestamp":1561373812474}
```


Data OpenStream API方法说明

连接信息

- Host:
中国站: sensecap-openstream.seeed.cn
国际站: 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> 使用您自己编写的ID (支持数字和小写字母)。
- Username: 格式为org-<Organization ID>, <Organization ID>可以在您的云平台“安全/API访问密钥”页面中获取 (参考快速入门)。
- Password: 在您的云平台“安全/API访问密钥”页面中获取“API访问密钥” (参考快速入门)。

发布和订阅模型

SenseCAP Data OpenStream 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subscribe方法, 您可以订阅并监听设备上上报的传感器遥测数据等消息。

消息主题

接收设备遥测数据

主题格式: /device_sensor_data/<OrgID>/<DeviceEUI>/<Channel>/<Reserved>/<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
Channel	一个设备上的物理接口, 用于连接传感器, 默认为1
Reserved	预留字段
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账号下的设备。而且<OrgID>字段是必填的。

消息主体

```
{
  "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在一起使用。

字段	描述
----	----

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

接收设备状态数据

主题格式：/device_status_event/<OrgID>/<DeviceEUI>/<Reserved>/<StatusID>

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
Reserved	预留字段
StatusID	状态值ID，具体设备状态ID清单详见附录

请依照设备状态ID列表订阅需要的StatusID，避免订阅到意料之外的ID

消息主体

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

字段	描述
value	状态值内容
timestamp	数据的时间戳，单位为毫秒

Sensecap Java SDK Quickstart

Summary:

This guide will guide you how to use Sensecap Java SDK

Overview

Prerequisite

If you do not have an account, please register for the SenseCAP Portal.

- China Station <https://sensecap.seeed.cn>
- Global Station <https://sensecap.seeed.cc>

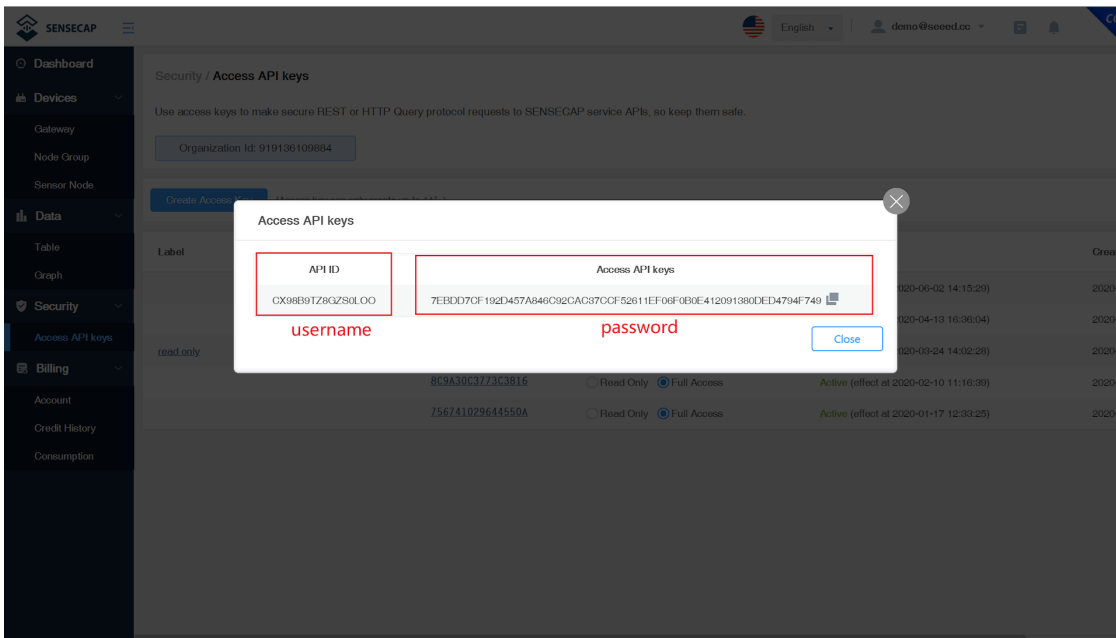
Note: LoRaWAN devices are used with Global Station

Get an Access Key

1. Login the SenseCAP Portal.
2. Navigate to “Security/Access API keys”
3. Click “Create Access Key”
4. Click “API ID”, and get the “API ID” and “Access API keys” after entering the password.

The screenshot shows the SenseCAP portal interface. On the left sidebar, the 'Security' menu is expanded, and 'Access API keys' is selected. The main content area shows the 'Security / Access API keys' page. It includes a warning about using access keys for REST or HTTP requests, the organization ID '919136109884', and a 'Create Access Key' button. Below this is a table of API keys.

Label	API ID	Access Level	API Status	Create Time
	CX98B91Z86ZS0L00	☐ Read Only ☒ Full Access	Active (effect at 2020-06-02 14:15:29)	2020-0
	5FAE881GZ0GUYJJ	☐ Read Only ☒ Full Access	Active (effect at 2020-04-13 16:36:04)	2020-0
read only	K00M81SYN91ZK1TG	☒ Read Only ☐ Full Access	Active (effect at 2020-03-24 14:02:28)	2020-0
	8C9A30C3773C3816	☐ Read Only ☒ Full Access	Active (effect at 2020-02-10 11:16:39)	2020-0
	756741029644550A	☐ Read Only ☒ Full Access	Active (effect at 2020-01-17 12:33:25)	2020-0



Maven coordinates

```
<dependency>
  <groupId>cc.seeed.sensecap-sdk</groupId>
  <artifactId>sensecap-java-sdk</artifactId>
  <version>1.0-RELEASE</version>
</dependency>
```

SenseCAPClient initialization

```
//client Configure
{
  String accessId = " ";
  String accessKey = "";
  int region = RegionType.SENSECAP_CC.getRegion();
  //SenseCAPClient senseCAPClient = new SenseCAPClientBuilder().buildConfig(accessId, accessKey, region)
  ;
  OpenApiConfig openApiConfig = new OpenApiConfig(accessId, accessKey, region);
  SenseCAPClient senseCAPClient = new SenseCAPClientBuilder().buildConfig(openApiConfig);
}
```

Organization

Get organizationId

```
public long getOrganizationId() throws BaseException;
```

Request parameters

None

Return values

Type	Description
long	organizationId

Request Example

```
public static void getOrganizationId() throws BaseException{
    long organizationId = senseCAPClient.getOrganizationManager()
        .getOrganizationId();
    System.out.println(organizationId);
}
```

Group

Create Group

```
public GroupInfo create(String groupName) throws BaseException;
```

Request parameters

Name	Type	Description
groupName	String	Group name

Return values

Type	Description
GroupInfo	Group information

Request Example

```
public static void createGroup() throws BaseException {
    GroupInfo result = senseCAPClient.getGroupManager().create("SDK-TEST");
    System.out.println(result);
}
```

Rename group

```
public void rename(String groupUUID, String groupName) throws BaseException;
```

Request parameters

Name	Type	Description
groupUUID	String	GroupUUID

Name	Type	Description
groupName	String	Group name

Return values

None

Request Example

```
public static void renameGroup() throws BaseException {
    senseCAPClient.getGroupManager().rename("groupUUID", "Sdk-TEST");
}
```

Remove group

```
public void remove(String groupUUID) throws BaseException;
```

Request parameters

Name	Type	Description
groupUUID	String	GroupUUID

Return values

None

Request Example

```
public static void renameGroup() throws BaseException {
    senseCAPClient.getGroupManager().remove("GroupUUID");
}
```

Get all group

```
public List<GroupInfo> getGroupList() throws BaseException;
```

Request parameters

None

Return values

Type	Description
list	Group information list

Request Example

```
public static void getGroupList() throws BaseException {
    GroupResult groupResult = senseCAPClient.getGroupManager().createGroupQuery()
        .build()
        .execute();
    List<GroupInfo> groupInfos = groupResult.toList();
    System.out.println(groupInfos.toString());
}
```

Device

Move device

```
public void moveDevices(String groupUUID, List<String> deviceEuis) throws BaseException;
```

Request parameters

Name	Type	Description
groupUUID	String	GroupUUID
deviceEuis	array	Device eui list

Return values

None

Request Example

```
public static void moveDeivces() throws BaseException {
    List<String> deviceEuis = Lists.newArrayList();
    deviceEuis.add(deviceEui);
    senseCAPClient.getDeviceManager().moveDevices("groupUUID", deviceEuis);
}
```

Get all devices under the group

```
public List<DeviceBaseInfo> getDeviceList(int deviceType, String groupUUID) throws BaseException;
```

Request parameters

Name	Type	Description
groupUUID	String	GroupUUID
deviceType	int	Device type 1: gateway ,2: node(default)

Return values

Type	Description
list	device base information list

Request Example

```
public static void getDeviceList() throws BaseException {
    DeviceResult deviceResult = senseCAPClient.getDeviceManager().createDeviceQuery()
        .deviceType(2)
        .groupUUID("")
        .build()
        .execute();
    List list = deviceResult.deviceList();
    System.out.println(list.toString());
}
```

Get device information

```
public List<DeviceInfo> getDeviceInfo(List<String> deviceEuis, int deviceType) throws BaseException;
```

Request parameters

Name	Type	Description
deviceEuis	String	Device eui list
deviceType	int	Device type 1: gateway ,2: node(default)

Return values

Type	Description
list	Device information list

Request Example

```
public static void getDeviceInfoList() throws BaseException {
    List<String> deviceEuis = Lists.newArrayList();
    deviceEuis.add(deviceEui);
    DeviceResult deviceResult = senseCAPClient.getDeviceManager().createDeviceQuery()
        .deviceEuis(deviceEuis)
        .deviceType(2)
        .build()
        .execute();
    List<DeviceInfo> deviceInfos = deviceResult.deviceInfos();
    System.out.println(deviceInfos.toString());
}
```

Get device channel information

```
public List<DeviceChannelInfo> getDeviceChannelList(List<String> deviceEuis) throws BaseException;
```

Request parameters

Name	Type	Description
------	------	-------------

Name	Type	Description
deviceEuis	String	device eui list

Return values

Type	Description
list	Device channel information list

Request Example

```
public static void getDeviceChannelList() throws BaseException {
    List<String> deviceEuis = Lists.newArrayList();
    deviceEuis.add(deviceEui);
    DeviceResult deviceResult = senseCAPClient.getDeviceManager().createDeviceQuery()
        .deviceEuis(deviceEuis)
        .build()
        .execute();
    List<DeviceChannelInfo> deviceChannelInfos = deviceResult.channelList();
    System.out.println(deviceChannelInfos.toString());
}
```

Get device running status

```
public List<DeviceStatusInfo> getDeviceRunningStatusList(List<String> deviceEuis) throws BaseException;
```

Request parameters

Name	Type	Description
deviceEuis	String	device eui list

Return values

Type	Description
list	device running status list

Request Example

```
public static void getDeviceRunningStatusList() throws BaseException {
    List<String> deviceEuis = Lists.newArrayList();
    deviceEuis.add(deviceEui);
    DeviceResult deviceResult = senseCAPClient.getDeviceManager().createDeviceQuery()
        .deviceEuis(deviceEuis)
        .build()
        .execute();
    List<DeviceStatusInfo> deviceStatusInfos = deviceResult.runningStatusList();
    System.out.println(deviceStatusInfos);
}
```

Bind device

```
public boolean bindDevice(String eui, String code, String deviceName, String groupUUID, String longitude, String latitude) throws BaseException;
```

Request parameters

Name	Type	Description
eui	String	device eui
code	String	device code
deviceName	String	device name
groupUUID	String	groupUUID
longitude	String	longitude
latitude	String	latitude

Return values

None

Request Example

```
public static void bindDevice() throws BaseException {  
    senseCAPClient.getDeviceManager().createBinder()  
        .eui("")  
        .code("")  
        .deviceName("")  
        .groupUUID("")  
        .longitude("")  
        .latitude("")  
        .build()  
        .bind();  
}
```

Delete device

```
public boolean deleteDevices(List<String> deviceEuis) throws BaseException;
```

Request parameters

Name	Type	Description
deviceEuis	String	device eui list

Return values

Type	Description
boolean	success or not

Request Example

Request Example

```
public static void deleteDevices() throws BaseException {
    List<String> deviceEuis = Lists.newArrayList();
    deviceEuis.add(deviceEui);
    boolean b = senseCAPClient.getDeviceManager().deleteDevices(deviceEuis);
}
```

Telemetry data

Get latest telemetry data

```
List<LatestTelemetryDataInfo> getLatestTelemetryData(String deviceEui, int channelIndex, int measurementId) throws BaseException;
```

Request parameters

Name	Type	Description
deviceEui	String	device eui
channelIndex	int	channel
measurementId	int	measurement id

Return values

Type	Description
list	latest telemetry data list

Request Example

```
public static void getLatestTelemetryData() throws BaseException {
    DataResult dataResult = senseCAPClient.getDataManager().createDataQuery()
        .deviceEui(deviceEui)
        .channelIndex(1)
        .measurementId(4099)
        .build()
        .execute();
    List<LatestTelemetryDataInfo> latestTelemetryDataInfos = dataResult.latestData();
    System.out.println(latestTelemetryDataInfos.toString());
}
```

Get historical telemetry data

```
List<TelemetryDataInfo> getTelemetryDataListCallback(String deviceEui, int channelIndex, int measurementId, int limit, long startTime, long endTime, TelemetryDataCallback var) throws BaseException;
```

Request parameters

Name	Type	Description
deviceEui	String	device eui
channelIndex	int	channel
measurementId	int	measurement id
limit	int	Number of queries
startTime	long	start time
endTime	long	end time
var	TelemetryDataCallback	Callback function

Parameter description

- [startTime>0,endTime>0] Get the historical data of the specified time range [-30d,now]
- (startTime=0,endTime>0] Get the default one day old to endTime historical data [-1d,endTime]
- [startTime>0,endTime=0] Get the historical data of the start time and subscribe to the latest news [startTime,-]
- (startTime=0,endTime=0) Subscribe to the latest news only [now,-]
- Service restart requires user to handle reconnection subscription by self

Return values

Type	Description
list	historical telemetry data list

Request Example

```
public static void getTelemetryDataListCallback() throws BaseException {
    DataResult dataResult = senseCAPClient.getDataManager().createDataQuery()
        .deviceEui(deviceEui)
        .channelIndex(1)
        //.measurementId(4100)
        .startTime(1605385871000L)
        //.endTime(1605496271000L)
        .limit(100)
        .callback(new TelemetryDataCallback() {
            @Override
            public void messageArrived(TelemetryDataResult var) throws BaseException {
                System.out.println("Test get mqtt messages: " + var.toString());
            }
        })
        .build()
        .execute();
    List<TelemetryDataInfo> list = dataResult.dataList();
    System.out.println(JSON.toJSONString(list));
}
```

Get device Line chart data(openAPI)

```
List<ChartPointDataInfo> getChartPointData(String deviceEui, int channelIndex, int measurementId, int interval, long startTime, long endTime) throws BaseException;
```

Explain

- Divide the large data segment into small data segments, then output the average value of each segment.
- The return data is up to one year and 250 points each measurement at most.If more than 250 points, it will automatically re-divide the time period to return 250 points.

Request parameters

Name	Type	Description
deviceEui	String	device eui
channelIndex	int	channel
measurementId	int	measurement id
interval	int	Interval minutes
startTime	long	start time Timestamp MS
endTime	long	end time Timestamp MS

Return values

Type	Description
list	device Line chart data list

Request Example

```
public static void getLatestTelemetryData() throws BaseException {
    DataResult dataResult = senseCAPClient.getDataManager().createDataQuery()
        .startTime(1605600000000L)
        .deviceEui(deviceEui)
        .build()
        .execute();
    List<ChartPointDataInfo> chartPointDataInfos = dataResult.chartPointData();
    System.out.println(JSON.toJSONString(chartPointDataInfos));
}
```

Get device Line chart data

```
List<ChartDataListInfo> getChartDataList(String deviceEui, int channelIndex, int measurementId, int interval, long startTime, long endTime) throws BaseException;
```

Explain

- Return all data points within one day (inclusive)
- If the time is greater than one day, the average value of data points in the **interval** range is returned
- **interval** should not be less than the minimum interval of data reporting

Request parameters

Name	Type	Description
deviceEui	String	device eui
channelIndex	int	channel

Name	Type	Description
measurementId	int	measurement id
interval	int	Interval minutes
startTime	long	start time Timestamp MS
endTime	long	end time Timestamp MS

Return values

Type	Description
list	device Line chart data list

Request Example

```
public static void getLatestTelemetryData() throws BaseException {
    DataResult<T> dataResult1 = senseCAPClient.getDataManager().createDataQuery()
        .startTime(1605600000000L)
        .deviceEui(deviceEui)
        .interval(30)
        .build()
        .execute();
    List<ChartDataListInfo> listInfoList = dataResult1.chartData();
    System.out.println(JSON.toJSONString(listInfoList));
}
```

传感器的类型表

传感器ID	传感器名称	测量值ID	测量值名称
1001	空气温湿度传感器	4098, 4097	空气湿度, 空气温度
1003	光照强度传感器	4099	光照
1004	二氧化碳传感器	4100	二氧化碳
1005	气压传感器	4101	气压
1006	土壤温湿度传感器	4102, 4103	土壤温度, 土壤湿度
1008	风向传感器	4104	风向
1009	风速传感器	4105	风速
1011	雨量传感器	4113	每小时降雨量
1013	超声波测距传感器	4115	距离
1014	浸液传感器	4116	浸液
1015	液位传感器	4117	液位
2001	五合一传感器(S500)	4097, 4098, 4101, 4104, 4105	空气温度, 空气湿度, 气压, 风向, 风速
2002	三合一传感器(S300)	4097, 4101, 4098	空气温度, 气压, 空气湿度
2003	四合一传感器(S400)	4097, 4098, 4101, 4099	空气温度, 空气湿度, 气压, 光照
2004	氨气温湿度传感器	4097, 4098, 4118	空气温度, 空气湿度, 氨气
2005	硫化氢温湿度传感器	4097, 4098, 4119	空气温度, 空气湿度, 硫化氢
2007	土壤水分温度传感器	4110, 4102	土壤体积含水量, 土壤温度
2009	涡轮流量计	4121, 4120	累计流量, 瞬时流量
2010	日照时数传感器	4126	日照时数
2011	太阳总辐射传感器	4127	太阳总辐射
2012	蒸发量传感器	4128	水面蒸发量
2013	光合有效辐射传感器	4129	光合有效辐射

2015	土壤热通量传感器	4125	土壤热通量
2016	土壤张力计	4133	土壤张力
2017	水质EC传感器	4135, 4134, 4124, 4123	总溶解固体, 盐度, 水质温度, 水质电导率
2018	叶面温湿度传感器	4137, 4136	叶面湿度, 叶面温度
2019	多层土壤温湿度传感器	4144, 4138, 4139, 4140, 4141, 4142, 4143, 4145	土壤温度-30cm, 土壤容积含水率-10cm, 土壤容积含水率-20cm, 土壤容积含水率-30cm, 土壤容积含水率-40cm, 土壤温度-10cm, 土壤温度-20cm, 土壤温度-40cm
201A	九合一传感器 (S900)	4147, 4146, 4097, 4098, 4099, 4101, 4104, 4105, 4113	PM10, PM2.5, 空气温度, 空气湿度, 光照, 气压, 风向, 风速, 每小时降雨量
100A	pH传感器	4106	pH
100B	光通量传感器	4107	光通量
100C	电导传感器	4108	土壤电导率
100D	溶解氧传感器	4109	溶解氧
100E	土壤含水量温度与电导率传感器	4108, 4102, 4110	土壤电导率, 土壤温度, 土壤体积含水量
200A	七合一传感器(S700)	4098, 4099, 4101, 4104, 4105, 4113, 4097	空气湿度, 光照, 气压, 风向, 风速, 每小时降雨量, 空气温度
200D	氧气传感器	4122	氧气浓度
200E	水质电导率传感器	4123	水质电导率

测量值信息表

测量值ID	测量值名称	数值范围	单位
4097	空气温度	-40~90	°C
4098	空气湿度	0~100	%RH
4099	光照	0~188000	Lux
4100	二氧化碳	0~10000	ppm
4101	气压	300~1100000	Pa
4102	土壤温度	-30~70	°C
4103	土壤湿度	0~100	%RH
4104	风向	0~360	°
4105	风速	0~60	m/s
4106	pH	0~14	PH
4107	光通量	0~5000	umol/m ² s
4108	土壤电导率	0~23	dS/m
4109	溶解氧	0~20	mg/L
4110	土壤体积含水量	0~100	%
4111	土壤电导率	0~23	dS/m
4112	土壤温度(土壤含水量温度电导传感器)	-40~60	°C
4113	每小时降雨量	0~240	mm/hour
4115	距离	28~250	cm
4116	浸液	true / false	
4117	液位	0~500	cm
4118	氨气	0~100	ppm
4119	硫化氢	0~100	ppm
4120	瞬时流量	0~65535	m ³ /h
4121	累计流量	0~6553599	m ³
4122	氧气浓度	0~25	%vol
4123	水质电导率	0~20000	us/cm
4124	水质温度	-40~80	°C
4125	土壤热通量	-500~500	W/m ²
4126	日照时数	0~24	h
4127	太阳总辐射	0~5000	W/m ²
4128	水面蒸发量	0~10000	mm
4129	光合有效辐射	0~5000	umol/m ² s

4130	加速度陀螺仪	0,0,0~x.xx,y.yy,z.zz	m/s
4131	响度	0~100	dB
4133	土壤张力	-100~0	KPA
4134	盐度	0~20000	mg/L
4135	总溶解固体	0~20000	mg/L
4136	叶面温度	0~100	°C
4137	叶面湿度	-40~85	%
4138	土壤容积含水率-10cm	0~100	%
4139	土壤容积含水率-20cm	0~100	%
4140	土壤容积含水率-30cm	0~100	%
4141	土壤容积含水率-40cm	0~100	%
4142	土壤温度-10cm	-30~70	°C
4143	土壤温度-20cm	-30~70	°C
4144	土壤温度-30cm	-30~70	°C
4145	土壤温度-40cm	-30~70	°C
4146	PM2.5	0~1000	μg/m3
4147	PM10	0~2000	μg/m3
4150	加速度x轴	-49.99~49.99	m/s
4151	加速度y轴	-49.99~49.99	m/s
4152	加速度z轴	-49.99~49.99	m/s
5100	开关状态	100~200	

设备状态ID表

SensorHub

Status ID	Status name
3000	battery
3009	Signal strength
3013	The onboard temperature
3017	Battery charging state(1:In the charging, 2:Full charged,3:No full charged)
3900	Data collection interval

LoraPP Node

Status ID	Status name
3000	battery,Once a day

LoraWan Node

Status ID	Status name
3000	battery,Once a day

错误码列表

code	msg	说明
10000	Framework analysis error	系统发生未知错误
11201	You have no options permission	没有操作权限
11202	Request parameters are invalid	请求参数无效
11203	Invalid device EUI	EUI格式错误
11317	Group not exist	分组不存在
11327	Group is exists	分组已存在
11330	Check eui error	设备EUI不存在
11334	Eui is binded	设备已被绑定