

IDEAS Chain

數據平台 API 說明文件

目錄

取得 JWT Token	3
裝置管理.....	4
新增裝置.....	4
刪除裝置.....	5
修改裝置.....	6
查詢裝置 by deviceId	6
查詢裝置 accessToken by deviceId	7
更新裝置 accessToken.....	8
查詢 Tenant 名下裝置.....	9
資料上傳.....	11
時間序列資料上傳(Telemetry upload).....	11
HTTP	11
MQTT.....	13
CoAP	13
屬性上傳(Attributes upload).....	15
HTTP	15
MQTT.....	16
CoAP	17
資料存取.....	19
存取時間序列資料(Request telemetry values)	19
HTTP	19
存取屬性數據(Request attribute values).....	22
HTTP	22
MQTT.....	22
Remote Procedure Call (RPC).....	25

取得 JWT Token

本平台使用 JWT 作為請求認證，欲連接的 API Header 中有包含 **\$YOUR_JWT_TOKEN** 時即代表需要您的 JWT token。

- Method：POST
- Url：

```
https://iiot.ideaschain.com.tw/api/auth/login
```

- Header：

```
Content-type:application/json  
Accept:application/json
```

- Parameters：

```
{  
  "username":"iiiagvgroup@gmail.com",  
  "password":"iii1234!"  
}
```

- Response Body:

```
{
  "token":
    "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJzeXNhZG1pbkBpaWkub3JnLnR3liwic2NvcGZlZjpbIiJFRlJFU0hfVE9LRU4iXSwidXNlcklkIjoiNWU3OTc2NjAtNDYxMi0xMWU3LWE5MTktOTJlYmNiNmJmZTMzliwiaXNqdWJsaWMiOmZhbHNlLCJpc3MiOiJ0aGluZ3Nib2FyZC5pbyslImp0aSI6IjA5NmMxMGNkLTlxMDYtNDE3ZC05ZjI5LTJhYmYzMmMzRjNyIj0E1NzM4MDYyNjN9.xHerm8m-_oiaEP6RpuR_R-gBvEVo9gHJjCMqD39x3UHyxgE-3st7vdr44747RyUCIR8IJhXlBPBccvm8aE5gbA",
  "refreshToken":
    "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJzeXNhZG1pbkBpaWkub3JnLnR3liwic2NvcGVzljpbIiJFRlJFU0hfVE9LRU4iXSwidXNlcklkIjoiNWU3OTc2NjAtNDYxMi0xMWU3LWE5MTktOTJlYmNiNmJmZTMzliwiaXNqdWJsaWMiOmZhbHNlLCJpc3MiOiJ0aGluZ3Nib2FyZC5pbyslImp0aSI6IjA5NmMxMGNkLTlxMDYtNDE3ZC05ZjI5LTJhYmYzMmMzRjNyIj0E1NzM4MDYyNjN9.xHerm8m-_oiaEP6RpuR_R-gBvEVo9gHJjCMqD39x3UHyxgE-3st7vdr44747RyUCIR8IJhXlBPBccvm8aE5gbA"
}
```

裝置管理

新增裝置

- Method : POST
- Url :

<https://iiot.ideaschain.com.tw/api/device>

- Header :

```
Content-type:application/json
Accept:application/json
X-Authorization:Bearer $YOUR_JWT_TOKEN
```

- Parameters :

```
{
  "name": "string",
  "type": "string"
}
```

- Response Body:

```
{
  "id": {
    "entityType": "DEVICE",
    "id": "88226eb0-0f36-11ea-b791-25f06e619e9c"
  },
  "createTime": 1574653747483,
  "additionalInfo": null,
  "tenantId": {
    "entityType": "TENANT",
    "id": "cc7c6730-b8e4-11e9-8864-4f072b21e9a2"
  },
  "customerId": {
    "entityType": "CUSTOMER",
    "id": "13814000-1dd2-11b2-8080-808080808080"
  },
  "name": "API_Test",
  "type": "api"
}
```

刪除裝置

- Method : DELETE
- Url :

```
https://iiot.ideaschain.com.tw/api/device/$deviceId
```

- Header :

```
Accept:application/json
X-Authorization:Bearer $YOUR_JWT_TOKEN
```

修改裝置

- Method : POST
- Url :

```
https://iiot.ideaschain.com.tw/api/device
```

- Header :

```
Content-type:application/json  
Accept:application/json  
X-Authorization:Bearer $YOUR_JWT_TOKEN
```

- Parameters :

```
{  
  "id": {  
    "entityType": "DEVICE",  
    "id": "$deviceId"  
  },  
  "name": "string",  
  "type": "string"  
}
```

查詢裝置 by deviceId

- Method : GET
- Url :

```
https://iiot.ideaschain.com.tw/api/device/$deviceId
```

- Header :

```
X-Authorization:Bearer $YOUR_JWT_TOKEN
```

- Response Body:

```
{
  "id": {
    "entityType": "DEVICE",
    "id": "88226eb0-0f36-11ea-b791-25f06e619e9c"
  },
  "createdTime": 1574653747483,
  "additionalInfo": null,
  "tenantId": {
    "entityType": "TENANT",
    "id": "cc7c6730-b8e4-11e9-8864-4f072b21e9a2"
  },
  "customerId": {
    "entityType": "CUSTOMER",
    "id": "13814000-1dd2-11b2-8080-808080808080"
  },
  "name": "API_Test",
  "type": "api"
}
```

查詢裝置 accessToken by deviceId

- Method : GET
- Url :

[https://iiot.ideaschain.com.tw/api/device/\\$deviceId/credentials](https://iiot.ideaschain.com.tw/api/device/$deviceId/credentials)

- Header :

X-Authorization:Bearer **\$YOUR_JWT_TOKEN**

- Response Body:

```
{
  "id": {
    "id": "8822bcd0-0f36-11ea-b791-25f06e619e9c"
  },
  "createdTime": 15746
53747485,
  "deviceId": {
    "entityType": "DEVICE",
    "id": "88226eb0-0f36-11ea-b791-25f06e619e9c"
  },
  "credentialsType": "ACCESS_TOKEN",
  "credentialsId": "n4Bp9pY93uLN18k3Q6F6",
  "credentialsValue": null
}
```

更新裝置 accessToken

- Method : POST
- Url :

<https://iiot.ideaschain.com.tw/api/device/credentials>

- Header :

Content-type:application/json
Accept:application/json
X-Authorization:Bearer **\$YOUR_JWT_TOKEN**

- Parameters :


```
{
  "id": {
    "id": "8822bcd0-0f36-11ea-b791-25f06e619e9c"
  },
  "deviceId": {
    "entityType": "DEVICE",
    "id": "$deviceId"
  },
  "credentialsType": "ACCESS_TOKEN",
  "credentialsId": "n4Bp9pY93uLN18k3Q6F6",
  "credentialsValue": null
}
```

查詢 Tenant 名下裝置

- Method : GET
- Url :

<https://iiot.ideaschain.com.tw/api/tenant/devices?limit=30&textSearch=>

- Header :

X-Authorization:Bearer **\$YOUR_JWT_TOKEN**

- Supported parameters :

type - Device type name
textSearch - Device name keyword search text
idOffset - Next page link parameter
textOffset - Next page link parameter

- Response Body:

```
{
  "data": [
    {
      "id": {
        "entityType": "DEVICE",
        "id": "1f592210-bcad-11e9-8864-4f072b21e9a2"
      },
      "createdTime": 1565578735281,
      "additionalInfo": null,
      "tenantId": {
        "entityType": "TENANT",
        "id": "cc7c6730-b8e4-11e9-8864-4f072b21e9a2"
      },
      "customerId": {
        "entityType": "CUSTOMER",
        "id": "13814000-1dd2-11b2-8080-808080808080"
      },
      "name": "KM_T_D",
      "type": "test"
    }
  ],
  "nextPageLink": {
    "limit": 1,
    "textSearch": "",
    "textSearchBound": null,
    "textOffset": "km_t_d",
    "idOffset": "1f592210-bcad-11e9-8864-4f072b21e9a2"
  },
  "hasNext": true
}
```

資料上傳

時間序列資料上傳(Telemetry upload)

適合彙集遙測數據，具有時間序列的值

HTTP

- Method : POST
- Url :

```
https://iiot.ideaschain.com.tw/api/v1/$ACCESS_TOKEN/telemetry
```

Note: \$ACCESS_TOKEN 可於各裝置管理認證中存取權杖取得

- Header :

```
Content-type:application/json
```

- Body :
 1. Upload data example

```
{"key1":"value1", "key2":"value2"}  
or  
[{"key1":"value1"}, {"key2":"value2"}]
```

2. Upload data with ts example

```
{"ts":1451649600512, "values":{"key1":"value1", "key2":"value2"}}
```

Note: ts should be a [unix timestamp](#) with milliseconds precision

- Server Response

```
200 OK  
400 Bad Request - Invalid URL, request parameters or body.  
401 Unauthorized - Invalid $ACCESS_TOKEN.  
404 Not Found - Resource not found.
```

- Sample Code
 1. cURL

```
curl --location --request POST
'https://iiot.ideaschain.com.tw/api/v1/$ACCESS_TOKEN/telemetry' \
--header 'Content-Type: application/json' \
--data-raw '{
    "key1": "value1",
    "key2": "value2"
}'
```

2. Nodejs

```
var request = require('request');
var options = {
  'method': 'POST',
  'url': 'https://iiot.ideaschain.com.tw/api/v1/$ACCESS_TOKEN/telemetry',
  // $ACCESS_TOKEN 可於各裝置管理認證中存取權杖取得
  'headers': {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({"key1":"value1","key2":"value2"})
};
request(options, function (error, response) {
  if (error) throw new Error(error);
  console.log(response.body);
});
```

3. Python

```
import requests

url = "https://iiot.ideaschain.com.tw/api/v1/$ACCESS_TOKEN/telemetry"

payload = "{\n    \"key1\": \"value1\", \n    \"key2\": \"value2\" \n}"
headers = {
  'Content-Type': 'application/json'
}

response = requests.request("POST", url, headers=headers, data = payload)

print(response.text.encode('utf8'))
```

MQTT

- Host : iiot.ideaschain.com.tw
- Port : 1883
- Topic :

v1/devices/me/telemetry

- User : **\$ACCESS_TOKEN**

Note: \$ACCESS_TOKEN 可於各裝置管理認證中存取權杖取得

- Message :

1. Upload data example

```
{"key1":"value1", "key2":"value2"}  
or  
[{"key1":"value1"}, {"key2":"value2"}]
```

2. Upload data with ts example

```
{"ts":1451649600512, "values":{"key1":"value1", "key2":"value2"}}
```

Note: ts should be a [unix timestamp](#) with milliseconds precision

- MQTT Connect

0x00 Connected - Successfully connected to MQTT server.

0x04 Connection Refused, bad user name or password - Username is empty.

0x05 Connection Refused, not authorized - Username contains invalid **\$ACCESS_TOKEN**.

- Sample Code

1. Mosquitto

```
mosquitto_pub -d -h "iiot.ideaschain.com.tw" -t "v1/devices/me/telemetry" -u  
"$ACCESS_TOKEN" -m '{"key1":"value1"}'
```

CoAP

- Method : POST
- Url :

```
coap://iiot.ideaschain.com.tw/api/v1/$ACCESS_TOKEN/telemetry
```

Note: \$ACCESS_TOKEN 可於各裝置管理認證中存取權杖取得

- Body :

1. Upload data example

```
{"key1":"value1", "key2":"value2"}  
or  
[{"key1":"value1"}, {"key2":"value2"}]
```

2. Upload data with ts example

```
{"ts":1451649600512, "values":{"key1":"value1", "key2":"value2"}}
```

Note: ts should be a [unix timestamp](#) with milliseconds precision

- Error codes

4.00 Bad Request - Invalid URL, request parameters or body.

4.01 Unauthorized - Invalid \$ACCESS_TOKEN.

4.04 Not Found - Resource not found.

- Sample Code

1. CoAP Client

```
echo -n '{"key1":"value1"}' | coap post  
coap://iiot.ideaschain.com.tw/api/v1/$ACCESS_TOKEN/telemetry
```

屬性上傳(Attributes upload)

適合存放不會變更或是變更頻率很低，只需追蹤最新且不需要保存變更前的值

HTTP

- Method : POST
- Url :

```
https://iiot.ideaschain.com.tw/api/v1/$ACCESS_TOKEN/attributes
```

Note: \$ACCESS_TOKEN 可於各裝置管理認證中存取權杖取得

- Header :

```
Content-type:application/json
```

- Body :
 1. Upload data example

```
{"attribute1":"value1", "attribute2":true, "attribute3":42.0, "attribute4":73}
```

- Server Response

```
200 OK
400 Bad Request - Invalid URL, request parameters or body.
401 Unauthorized - Invalid $ACCESS_TOKEN.
404 Not Found - Resource not found.
```

- Sample Code

1. cURL

```
curl --location --request POST
'https://iiot.ideaschain.com.tw/api/v1/$ACCESS_TOKEN/attributes' \
--header 'Content-Type: application/json' \
--data-raw '{
    "attribute1": "value1",
    "attribute2": true,
    "attribute3": 42
}'
```

2. Nodejs

```
curl --location --request POST
'https://iiot.ideaschain.com.tw/api/v1/$ACCESS_TOKEN/attributes' \
--header 'Content-Type: application/json' \
--data-raw '{
    "attribute1": "value1",
    "attribute2": true,
    "attribute3": 42
}'
```

3. Python

```
import requests

url = "https://iiot.ideaschain.com.tw/api/v1/$ACCESS_TOKEN/attributes"

payload = "{\n    \"attribute1\": \"value1\", \n    \"attribute2\": true,\n    \"attribute3\": 42\n}"
headers = {
    'Content-Type': 'application/json'
}

response = requests.request("POST", url, headers=headers, data = payload)

print(response.text.encode('utf8'))
```

MQTT

- Host : iiot.ideaschain.com.tw
- Port : 1883
- Topic :

```
v1/devices/me/attributes
```

- User : **\$ACCESS_TOKEN**

Note: \$ACCESS_TOKEN 可於各裝置管理認證中存取權杖取得

- Message :
 1. Upload data example


```
{"key1":"value1", "key2":"value2"}  
or  
[{"key1":"value1"}, {"key2":"value2"}]
```

2. Upload data with ts example

```
{"ts":1451649600512, "values":{"key1":"value1", "key2":"value2"}}
```

Note: ts should be a [unix timestamp](#) with milliseconds precision

- MQTT Connect

0x00 Connected - Successfully connected to MQTT server.
0x04 Connection Refused, bad user name or password - Username is empty.
0x05 Connection Refused, not authorized - Username contains invalid **\$ACCESS_TOKEN**.

- Sample Code

1. Mosquitto

```
mosquitto_pub -d -h "iiot.ideaschain.com.tw" -t "v1/devices/me/attributes" -u  
"$ACCESS_TOKEN" -m '{"attribute1":true}'
```

CoAP

- Method : POST
- Url :

```
coap://iiot.ideaschain.com.tw/api/v1/$ACCESS_TOKEN/telemetry
```

Note: \$ACCESS_TOKEN 可於各裝置管理認證中存取權杖取得

- Body :

1. Upload data example

```
{"key1":"value1", "key2":"value2"}  
or  
[{"key1":"value1"}, {"key2":"value2"}]
```

2. Upload data with ts example

```
{"ts":1451649600512, "values":{"key1":"value1", "key2":"value2"}}
```

Note: ts should be a [unix timestamp](#) with milliseconds precision

- Error codes

4.00 Bad Request - Invalid URL, request parameters or body.

4.01 Unauthorized - Invalid **\$ACCESS_TOKEN**.

4.04 Not Found - Resource not found.

- Sample Code

1. CoAP Client

```
echo -n '{"attribute1":true}' | coap post  
coap://iiot.ideaschain.com.tw/api/v1/$ACCESS_TOKEN/attributes
```

資料存取

存取時間序列資料(Request telemetry values)

HTTP

- Method : GET
- Url :
 1. 取得最新數值

```
https://iiot.ideaschain.com.tw/api/plugins/telemetry/DEVICE/$deviceId/values/timeseries?keys=key1,key2,key3
```

2. 取得時間區間內數值

```
https://iiot.ideaschain.com.tw/api/plugins/telemetry/DEVICE/$deviceId/values/timeseries?keys=key1,key2,key3&startTs=1479735870785&endTs=1479735871858&limit=100
```

- Header :

```
Content-type:application/json  
X-Authorization:Bearer $YOUR_JWT_TOKEN
```

- Supported parameters :

keys - comma separated list of telemetry keys to fetch.
startTs - unix timestamp that identifies start of the interval in milliseconds.
endTs - unix timestamp that identifies end of the interval in milliseconds.
limit - the max amount of data points to return or intervals to process.

- Sample Code

cURL

1. 取得最新數值

```
curl --location --request GET
'https://iiot.ideaschain.com.tw/api/plugins/telemetry/DEVICE/611ab430-78c2-
11ea-8281-ef3759c8954f/values/timeseries?keys=air_quality,humidity' \
--header 'Content-type: application/json' \
--header 'X-Authorization: Bearer
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0ZXN0QGlpaS5vcmcudHciLCJzY29wZXMiOiSiVE
VOQU5UX0FETUIOI0sInVzZXJJZCI6IjcyNjUxZTQwLTZmMTQzMTEFIYS05NzljLWFKZT
BkMGFkZDZmYyIsImVuYWJsZWQiOiOnRydWUslmlzUHVibGljIjpmYWxzZSwidGVuYW
50SWQioiI2MwY1ODFkMCO2ZjE0LTExZWEtOTc5Yy1hZGUwZDBhZGQ2ZmMiLCJjd
XN0b21lcklkIjoMTM4MTQwMDAtMWRkMi0xMWlyLTgwODAtODA4MDgwODA4
MDgwliwiaXNzIjoiaGhpYmVzYm9hcmQuaW8iLCJpYXQiOiE1ODczNjcyMzYsImV4cC
I6MTU4NzQ1MzYzNn0.u8DtZc7cVNJpvVBbRsyFS_bT6gQy9a7f1jurxuaGrI3VVFktW
0q0E2BWPY27irbi9dF_sQXE1RseGJjv6U9zQg'
```

2. 取得時間區間內數值

```
curl --location --request GET
'https://iiot.ideaschain.com.tw/api/plugins/telemetry/DEVICE/611ab430-78c2-
11ea-8281-
ef3759c8954f/values/timeseries?keys=air_quality,humidity&startTs=1587006116
000&endTs=1587009716000&interval=60000&limit=1000&agg=AVG' \
--header 'Content-type: application/json' \
--header 'X-Authorization: Bearer
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0ZXN0QGlpaS5vcmcudHciLCJzY29wZXMiOiSiVE
VOQU5UX0FETUIOI0sInVzZXJJZCI6IjcyNjUxZTQwLTZmMTQzMTEFIYS05NzljLWFKZT
BkMGFkZDZmYyIsImVuYWJsZWQiOiOnRydWUslmlzUHVibGljIjpmYWxzZSwidGVuYW
50SWQioiI2MwY1ODFkMCO2ZjE0LTExZWEtOTc5Yy1hZGUwZDBhZGQ2ZmMiLCJjd
XN0b21lcklkIjoMTM4MTQwMDAtMWRkMi0xMWlyLTgwODAtODA4MDgwODA4
MDgwliwiaXNzIjoiaGhpYmVzYm9hcmQuaW8iLCJpYXQiOiE1ODczNjcyMzYsImV4cC
I6MTU4NzQ1MzYzNn0.u8DtZc7cVNJpvVBbRsyFS_bT6gQy9a7f1jurxuaGrI3VVFktW
0q0E2BWPY27irbi9dF_sQXE1RseGJjv6U9zQg'
```

- Response Body :

```
{
  "air_quality": [
    {
      "ts": 1587009716734,
      "value": "86"
    }
  ],
  "humidity": [
    {
      "ts": 1587009153084,
      "value": "60.86"
    }
  ]
}
```

存取屬性數據(Request attribute values)

HTTP

- Method : GET
- Url :

```
https://iiot.ideaschain.com.tw/api/plugins/telemetry/DEVICE/$deviceId/values/attributes
```

- Header :

```
Content-type:application/json  
X-Authorization:Bearer $YOUR_JWT_TOKEN
```

- Response Body:

```
[  
  {  
    "lastUpdateTs": 1560237162478,  
    "key": "xPos",  
    "value": 463  
  },  
  {  
    "lastUpdateTs": 1560237166428,  
    "key": "yPos",  
    "value": 1200  
  }  
]
```

MQTT

- Host : iiot.ideaschain.com.tw
- Port : 1883
- Subscribe Topic :

```
v1/devices/me/attributes/response/+
```

- Publish Topic:

v1/devices/me/attributes/request/1

Note: Client need to subscribe topic first then send PUBLISH message to the publish topic.

- User : **\$ACCESS_TOKEN**

Note: \$ACCESS_TOKEN 可於各裝置管理認證中存取權杖取得

- Sample Code

Pure command-line examples are not available because **subscribe and publish need to happen in the same mqtt session.**

1. Nodejs

```
var mqtt = require('mqtt')
var client = mqtt.connect('mqtt://iiot.ideaschain.com.tw',{
  username: $ACCESS_TOKEN
})

client.on('connect', function () {
  console.log('connected')
  client.subscribe('v1/devices/me/attributes/response/+')
  client.publish('v1/devices/me/attributes/request/1',
    '{"clientKeys":"attribute1,attribute2", "sharedKeys":"shared1,shared2"}')
})

client.on('message', function (topic, message) {
  console.log('response.topic: ' + topic)
  console.log('response.body: ' + message.toString())
  client.end()
})
```

2. AT COMMAND

```
AT+Qrst=1
```

```
AT+QMTOPEN=0,"iiot.ideaschain.com.tw", 1883
```

```
AT+QMTCONN=0,"fZ2lsCcSGFYohsIJbPcN","fZ2lsCcSGFYohsIJbPcN"
```

```
AT+QMTSUB=0,1,"v1/devices/me/attributes/response/+",1
```

```
AT+QMTPUB=0,0,0,0,"v1/devices/me/attributes/request/1","{}"
```

```
AT+QMTPUB=0,0,0,0,"v1/devices/me/attributes/request/1","{"clientKeys":"music"}"
```

Note: MQTT Broker that supports QoS levels 0 (at most once) and 1 (at least once) and a set of predefined topics.

Remote Procedure Call (RPC)

- Host : iiot.ideaschain.com.tw
- Port : 1883
- Subscribe Topic :

```
v1/devices/me/rpc/request/+
```

訂閱後，客戶端將以下列形式接收相對應 Topic 傳送之指令

```
v1/devices/me/rpc/request/$request_id
```

Note: **\$request_id** 為整數識別值

- Sample Code

Pure command-line examples are not available because **subscribe and publish need to happen in the same mqtt session.**

1. Nodejs

```
var mqtt = require('mqtt')
var client = mqtt.connect('mqtt://iiot.ideaschain.com.tw', {
  username: "1z7ij659g8kf9bPjRjEO", // device access token (存取權杖)
  port: 1883,
  clientId: '8deaeaf0-856b-11eb-8e26-2532a0ef1bf0awewewq', // MQTT client
  ID: it's better to use unique id.
})
client.on('connect', function () {
  console.log('connected')

  client.subscribe('v1/devices/me/rpc/response/+', function (err, granted) {
    console.log(err, granted)
  })

  client.subscribe('v1/devices/me/rpc/request/+', function (err, granted) {
    console.log(err, granted)
  })
})

client.on('message', function (topic, message) {
  console.log('message topic: ' + topic)
  console.log('message body: ' + message.toString())
})
```

```
var rpcReq = 'v1/devices/me/rpc/request/'
var rpcResp = 'v1/devices/me/rpc/response/'

if (topic.startsWith(rpcReq)) {
  var reqId = topic.substr(rpcReq.length);
  var data = JSON.parse(message);
  var resp;
  // "method": "setValue"
  if (data.method == 'setValue') {
    resp = { value: true }
  } else if (data.method == 'getValue') {
    resp = { value: true }
  }
  else {
    console.log(data)
  }
  console.log('send to tb:' + rpcResp + reqId)
  if (!resp)
    resp = { data: 'abcde' }
  resp = JSON.stringify(resp);
  client.publish(rpcResp + reqId, resp);
}
// client.end()
})
client.on('disconnect', function (packet) {
  console.log('disconnect')
  client.end()
})
})
```