

# Thing+ Interworking Guide with HTTPS and OAuth2 - ENGLISH

## Contents

---

1. [Overview](#)
2. [Preparation](#)
3. [Create OAuth Client](#)
4. [Access Token Acquisition](#)
5. [Gateway Registration](#)
6. [Device Registration](#)
7. [Sensor Registration](#)
8. [Status Transmission](#)
9. [Sensor Value Transmission](#)

## 1. Overview

---

This document describes how to register gateways and sensors and send sensor values using the privileges of a user who can register the gateway, such as a service administrator or site administrator.

- Common gateway uses API key but it is not appropriate to apply the API key every time using the Thing+ Portal. It is applied when server or native app that can handle personal information such as user ID is linked with Thing+.
- Obtain authority using OAuth2, and register and transfer data and values using HTTPS.
- Actuators can not be linked using this method.
- If the account of the user who provided the privilege at the time of registration is deleted from Thing +, the privileges will disappear and your code will stop working.
- Acquired privileges are retained for 90 days, so you will need to reissue your access token when your due date expires.

## 2. Preparation

---

- Requires a tool that can call the HTTPS API to call the OAuth Client enrollment API.
  - [Google Chrome](#): Use this to sign in to Thing+ Portal.
  - [Postman](#): A Google Chrome App you can use to call the desired HTTPS API.
  - [Postman Interceptor](#): A Google Chrome extension that allows Postman to share cookies created when logged in to the Thing+ Portal.
  - Even if you do not use the above tool, you can still do so by using a tool that allows you to call the

HTTPS POST API while logged in from the Thing+ Portal.

- Please refer [Thing+ Support site](#).
- For more information about the APIs used, see [Thing+ API Reference](#).

### 3. Create OAuth Client

---

1. On **Chrome** , go to Thing+ Portal and log in with your service administrator account.
2. Execute **Postman** , set **Postman Interceptor** **on** , and then call the following API.

- URL: `https://api.sandbox.thingplus.net/v2/authClients`
- Method: POST
- Content-Type: `application/json`
- Body:

- Example

```
{
  "name": "Test Client for Daliworks",
  "reqId": "testClientId",
  "clientSecret": "testClientPwd12!@",
  "scopes": ["gateway", "site-read"]
}
```

- name: Auth client name. You can freely enter the name.
  - reqId: The ID of the auth client to be used to obtain the access token. Enter the value you have set.
  - clientSecret: This is used to get the access token from the auth client's secret value. Enter the value you have set.
  - scopes: List the permissions to grant to the auth client. See [link](#) for the values you can enter in scopes.
- Response
    - Example

```
{
  "statusCode": 201,
  "message": "Created",
  "data": {
    "name": "Test Client for Daliworks",
    "clientSecret": "testClientPwd12!@",
    "scopes": [
      "gateway",
      "site-read"
    ],
    "_user": "1",
    "mtime": 1495422902432,
    "ctime": 1495422902432,
    "id": "testClientId"
  }
}
```

## 4. Access Token Acquisition

---

1. Obtain an access token from the application using the following APIs:

- URL: <https://api.sandbox.thingplus.net/v2/oauth2/token>
- Method: POST
- Body:

- Example

```
{
  "grant_type": "password",
  "client_id": "testClientId",
  "client_secret": "testClientPwd12!@",
  "username": "serviceadmin",
  "password": "0b54b2a7b72f1efeb2c86885c3247787"
}
```

- grant\_type: Enter the string "password" as it is. (**This should not be your user password, please be careful.**)
- client\_id: Enter the ID value of the auth client generated by the /v2/authClient API.
- clientSecret: Enter the secret value of the auth client generated by the /v2/authClient API.
- username: Enter the ID of the user(usually the service administrator) who logged in to Thing + Portal when creating the auth client.
- password: Enter the md5 hash value of the user password that was logged into Thing +

Portal when creating the auth client.

- On OSX or Linux, use the following command to get the md5 hash value.

- OSX

```
$ echo -n your_password | md5  
0b54b2a7b72f1efeb2c86885c3247787
```

- Linux

```
$ echo -n your_password | md5sum  
0b54b2a7b72f1efeb2c86885c3247787  -
```

- In the environment where the above command does not work, you can get the md5 hash value of the password by using the md5 hash generator on the Internet, but be careful about security.
- [JavaScript-MD5](#) is also available.

- Response

- Example

```
{  
  "access_token": "2yJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiIxIiwiaW52xpZW50SWQiOiJ0ZXN0Q2xpZW50SWQiLCJpYXQiOiJlOTU0MzYwMzksImV4cCI6MTUwMzIxMjAzOX0.dQ65zRCgRml96fTc8CDnExAukrFPSLd7NzDlUkf4eYk",  
  "token_type": "Bearer"  
}
```

2. Use the acquired access token in the following way when calling the API.

- Add an `Authorization` field to the Header and a value of `token_type` and `access_token` in a single space.
- `curl` example

```
$ curl -H "Authorization: Bearer 2yJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiIxIiwiaW52xpZW50SWQiOiJ0ZXN0Q2xpZW50SWQiLCJpYXQiOiJlOTU0MzYwMzksImV4cCI6MTUwMzIxMjAzOX0.dQ65zRCgRml96fTc8CDnExAukrFPSLd7NzDlUkf4eYk" https://api.sandbox.thingplus.net/v2/gateways
```

## 5. Gateway Registration

1. Obtaining a site ID

- URL: <https://api.sandbox.thingplus.net/v2/sites>
- Method: GET
- Response:

- Example

```
{
  "statusCode": 200,
  "message": "OK",
  "data": [
    {
      "billing": "site:1",
      "code": "iot-service",
      "ctime": "1431416762406",
      "_service": "1",
      "billingReserve": "site:1:reserve",
      "billingLimit": "site:1:limit",
      "name": "default",
      "id": "1",
      "billingMeter": "site:1:meter",
      "mtime": "1431416762406",
      "billingCurrent": "site:1:current"
    }
  ]
}
```

- A list of sites is returned in the `data` of the response. (Usually 1, but if you create multiple sites, multiple site data will be returned.)
- Apply the ID of the site to which you want to register the gateway into the gateway registration API. (In the example, `"id": "1"` is the field corresponding to site ID.)

## 2. Obtaining the gateway model ID

- URL: <https://api.sandbox.thingplus.net/v2/gatewayModels/52>
- Method: GET
- Response:

- Example

```
{
  "statusCode": 200,
  "message": "OK",
  "data": {
    "reportInterval": "300000",
    "displayName": "Open API Gateway",
    "id": "52",
  }
}
```

```
"model": "open-api-gateway-v1.0",
"deviceModels": [
  {
    "id": "open-api-device-v1.0",
    "displayName": "Open API Device",
    "idTemplate": "{gatewayId}-{deviceAddress}",
    "discoverable": "y",
    "sensors": [
      {
        "network": "daliworks",
        "driverName": "openApiSensor",
        "model": "openApiTemp",
        "type": "temperature",
        "category": "sensor"
      },
      {
        "network": "daliworks",
        "driverName": "openApiSensor",
        "model": "openApiHumi",
        "type": "humidity",
        "category": "sensor"
      },
      ...
      {
        "network": "daliworks",
        "driverName": "openApiSensor",
        "model": "openApiConductivity",
        "type": "conductivity",
        "category": "sensor"
      }
    ],
    "max": 99
  }
],
"gatewayIdConfig": "n",
"mtime": "1495609234518",
"idFormat": "uuid",
"vendor": "OPEN API",
"ctime": "1495609234518",
"deviceMgmt": {
  "reportInterval": {
    "show": "y",
    "change": "y"
  },
  "DM": {
    "poweroff": {
```

```

        "support": "n"
    },
    "reboot": {
        "support": "n"
    },
    "restart": {
        "support": "n"
    },
    "swUpdate": {
        "support": "n"
    },
    "swInfo": {
        "support": "n"
    }
}
}
}
}
}

```

- The gateway model defines the characteristics of the gateway to be registered. What type of gateway ID to use, what devices and sensors to register, and so on.
- To register the gateway using the method described in this document, use the gateway model ID `52` (Open API Gateway). If you use a different gateway model, you can substitute `52` in the URL when calling this API by substituting another ID.

### 3. Creating a gateway ID

- For `Open API Gateway`, UUID is used as gateway ID.
- Obtain the UUID to be used for the ID of the newly registered gateway.
  - In case of OSX, it can be generated with `uuidgen` command.
  - In case of Linux(Ubuntu), it can be generated with `uuid` command after installing `uuid` package.
  - UUID can be obtained in various ways besides this method.
- Remove the `-` from the UUID obtained above, and change it to lowercase if it has uppercase characters to use as the gateway ID.

```

$ uuidgen | tr -d - | tr [:upper:] [:lower:]
366d685f93f5477a8d29e8c45bae0a31

```

### 4. Registering the gateway

- URL: `https://api.sandbox.thingplus.net/v2/registerGateway`
- Method: POST
- Body:

- Example

```
{
  "reqId": "87cd2a6e407511e7922eb724f8803770",
  "params": {
    "siteId": "1",
    "model": "52",
    "name": "Open Gateway 1"
  }
}
```

- reqId: Use the UUID generated above.
- siteId: Use the site ID obtained above.
- model: Use the gateway model ID obtained above.
- name: You can freely input the name of the gateway.
- For options other than those specified, see [Thing+ API Reference](#).

- Response

- Example

```
{
  "statusCode": 201,
  "message": "Created",
  "data": {
    "name": "Open Gateway 1",
    "model": "52",
    "_site": "1",
    "_service": "1",
    "reportInterval": "300000",
    "mtime": 1495626410619,
    "ctime": 1495626410619,
    "tree": "87cd2a6e407511e7922eb724f8803770",
    "id": "87cd2a6e407511e7922eb724f8803770"
  }
}
```

- reportInterval: This is the `transmission period`, which is the period(msec) at which the gateway transmits the measured value from the sensor to Thing+. This value can be modified in Thing+ Portal's `Gateway Management`.

## 6. Device Registration

---



## 1. Registering a device

- URL: `https://api.sandbox.thingplus.net/v2/gateways/{owner}/devices`
  - `{owner}`: Enter the ID of the gateway to which the device is connected. Use the gateway ID registered above.
- Method: POST
- Body:

- Example

```
{
  "reqId": "366d685f93f5477a8d29e8c45bae0a31",
  "name": "Open Gateway Device 1",
  "model": "open-api-device-v1.0"
}
```

- `reqId`: Generate a device ID using UUID and use it.
- `name`: You can freely input the name of the device.
- `model`: Use the device model ID in `deviceModels` when querying the gateway model. To register a device using the method described in this document, use the device model ID `open-api-device-v1.0`. If another device model is registered in the gateway model, you can substitute another device model ID when calling this API.
- For options other than those specified, see [Thing+ API Reference](#).

- Response

- Example

```
{
  "statusCode": 201,
  "message": "Created",
  "data": {
    "name": "Open Gateway 1",
    "model": "52",
    "_site": "1",
    "_service": "1",
    "reportInterval": "300000",
    "mtime": 1495626410619,
    "ctime": 1495626410619,
    "tree": "87cd2a6e407511e7922eb724f8803770",
    "id": "87cd2a6e407511e7922eb724f8803770"
  }
}
```

# 7. Sensor Registration

---

## 1. Registering a sensor

- URL: `https://api.sandbox.thingplus.net/v2/gateways/{owner}/sensors`
  - `{owner}`: Enter the ID of the gateway to which the sensor is connected. Use the gateway ID registered above.
- Method: POST
- Body:

- Example

```
{
  "reqId": "d815ba2f84eb490b8e68d9dd744da397",
  "name": "Temperature Sensor",
  "type": "temperature",
  "driverName": "openApiSensor",
  "model": "openApiTemp",
  "category": "sensor",
  "deviceId": "366d685f93f5477a8d29e8c45bae0a31"
}
```

- `reqId`: Generate a sensor ID using UUID and use it.
  - `name`: You can freely input the name for the sensor.
  - `type`: Use the sensor type value, `type` in `deviceModels.sensors` when querying the gateway model.
  - `driverName`: Use the value, `driverName` corresponding to the `type` in `deviceModels.sensors` when querying the gateway model.
  - `model`: Use the value, `model` corresponding to the `type` in `deviceModels.sensors` when querying the gateway model.
  - `category`: Enter the string "sensor" as it is. Currently, `actuator` is not supported in HTTPS mode.
  - `deviceId`: Enter the device ID to which the sensor is connected.
  - For options other than those specified, see [Thing+ API Reference](#).
- Response
    - Example
-

```
{
  "statusCode": 201,
  "message": "Created",
  "data": {
    "name": "Temperature Sensor",
    "type": "temperature",
    "driverName": "openApiSensor",
    "model": "openApiTemp",
    "category": "sensor",
    "deviceId": "366d685f93f5477a8d29e8c45bae0a31",
    "owner": "87cd2a6e407511e7922eb724f8803770",
    "mtime": 1495681664335,
    "ctime": 1495681664335,
    "id": "d815ba2f84eb490b8e68d9dd744da397"
  }
}
```

## 8. Status Transmission

---

### 1. Transferring the gateway status

- URL: <https://api.sandbox.thingplus.net/v2/gateways/{id}/status>
  - {id}: Enter the ID of the gateway. Use the gateway ID registered above.
- Method: PUT
- Body:
  - Example

```
{
  "validDuration": 450,
  "value": "on"
}
```

- validDuration: The duration(in seconds) for which `status` is valid. Typically you use 1.5 times the `reportInterval` .
  - value: The status of the gateway. You must send one 'on', 'off', or 'err'.
  - Response
    - Example
-

```
{
  "statusCode": 200,
  "message": "OK",
  "data": {
    "type": "status",
    "value": "on",
    "srcDbType": "gateway",
    "time": "1495703469608",
    "expireAt": "1495703919608",
    "vtime": "1495703469608",
    "status": "87cd2a6e407511e7922eb724f8803770",
    "owner": "87cd2a6e407511e7922eb724f8803770",
    "gateway": "87cd2a6e407511e7922eb724f8803770",
    "mtime": "1495703469610",
    "ctime": "1495683245619",
    "id": "status.gateway.pCMjRP"
  }
}
```

- `expireAt`: Indicates how long the `status` is valid, in Unix time in msec.
- The gateway status must be sent once per `reportInterval`.
- Even if the status of the gateway is `on`, after the `expireAt` time, you can not know what it is.

## 2. Transferring the sensor status

- URL: `https://api.sandbox.thingplus.net/v2/gateways/{owner}/sensors/{id}/status`
  - `{owner}`: Enter the ID of the gateway to which the sensor is connected. Use the gateway ID registered above.
  - `{id}`: Enter the sensor ID. Use the sensor ID registered above.
- Method: PUT
- Body:
  - Example

```
{
  "validDuration": 450,
  "value": "on"
}
```

- `validDuration`: The duration(in seconds) for which `status` is valid. Typically you use 1.5 times the `reportInterval`.
- `value`: The status of the sensor. You must send one 'on', 'off', or 'err'.

- Response

- Example

```
{
  "statusCode": 200,
  "message": "OK",
  "data": {
    "type": "status",
    "value": "on",
    "srcType": "temperature",
    "srcCategory": "sensor",
    "srcDbType": "sensor",
    "time": "1495715734508",
    "expireAt": "1495716184508",
    "vtime": "1495715720489",
    "status": "d815ba2f84eb490b8e68d9dd744da397",
    "owner": "87cd2a6e407511e7922eb724f8803770",
    "sensor": "d815ba2f84eb490b8e68d9dd744da397",
    "mtime": "1495715734508",
    "ctime": "1495715720489",
    "id": "status.sensor.0uRmI2"
  }
}
```

- expireAt: Indicates how long the `status` is valid, in Unix time in msec.

- The sensor status must be sent once per `reportInterval` .

- Even if the status of the sensor is `on` , after the expireAt time, you can not know what it is.

## 9. Sensor Value Transmission

---

### 1. Transferring the sensor value

- URL: <https://api.sandbox.thingplus.net/v2/gateways/{owner}/sensors/{id}/series>
  - {owner}: Enter the ID of the gateway to which the sensor is connected. Use the gateway ID registered above.
  - {id}: Enter the sensor ID. Use the sensor ID registered above.
- Method: PUT
- Body:
  - Example
    - In case that a sensor value is transmitted:

```
{
  "value": "36.5",
  "time": 1495715734508
}
```

- In case that multiple sensor values are transmitted:

```
[
  {
    "value": "36.5",
    "time": 1495716734508
  },
  {
    "value": "36.3",
    "time": 1495717184508
  }
]
```

- value: The sensor value. You must enter a string or object value and it must be proper to the sensor type. However, if the sensor value is numeric, it is sent as a string.
  - time: The time at which the sensor value was measured, in Unix time in msec.
  - If you want to send multiple sensor values, you can send them in an array.
- Response
    - Example
      - In case that a sensor value is transmitted:

```
{
  "statusCode": 200,
  "message": "OK",
  "data": {
    "name": "Temperature Sensor",
    "type": "temperature",
    "driverName": "openApiSensor",
    "model": "openApiTemp",
    "category": "sensor",
    "deviceId": "366d685f93f5477a8d29e8c45bae0a31",
    "owner": "87cd2a6e407511e7922eb724f8803770",
    "mtime": "1495717726517",
    "ctime": "1495681664335",
    "status": "status.sensor.0uRmI2",
    "series": "series.sensor.tOZhgI",
    "id": "d815ba2f84eb490b8e68d9dd744da397"
  }
}
```

- In case that multiple sensor values are transmitted:

```
{
  "statusCode": 200,
  "message": "OK",
  "data": {
    "type": "series",
    "srcType": "temperature",
    "seriesPack": [
      {
        "value": "36.5",
        "time": 1495718222169
      },
      {
        "value": "36.3",
        "time": 1495718672169
      }
    ],
    "mtime": "1495718691465",
    "ctime": "1495717726516",
    "time": "1495718672169",
    "value": "36.3",
    "owner": "87cd2a6e407511e7922eb724f8803770",
    "series": "d815ba2f84eb490b8e68d9dd744da397",
    "sensor": "d815ba2f84eb490b8e68d9dd744da397",
    "srcCategory": "sensor",
    "srcDbType": "sensor",
    "id": "series.sensor.tOZhgI"
  }
}
```