

HTTPS와 OAuth2를 이용한 Thing+ 연동 가이드

목차

1. [개요](#)
2. [준비 사항](#)
3. [OAuth Client 생성](#)
4. [Access Token 획득](#)
5. [Gateway 등록](#)
6. [Device 등록](#)
7. [Sensor 등록](#)
8. [Status 전송](#)
9. [Sensor 값 전송](#)

1. 개요

이 문서는 서비스 관리자나 사이트 관리자처럼 게이트웨이를 등록할 수 있는 사용자의 권한으로 게이트웨이와 센서를 등록하고 센서값을 전송하는 방법을 설명한 문서입니다.

- 일반적인 게이트웨이에서는 API key를 사용하지만, API key를 매번 Thing+ Portal에서 발급받아 적용하는 것이 적절하지 않고, 사용자의 ID와 같은 개인정보를 취급할 수 있는 server나 native app이 Thing+에 연동할 때 적용하는 방식입니다.
- OAuth2를 이용하여 권한을 획득하고, HTTPS를 이용하여 등록, 전송을 수행합니다.
- Actuator는 이 방식을 이용하여 연동할 수 없습니다.
- 등록할 때 권한을 제공한 사용자의 계정이 Thing+에서 삭제되면 권한이 사라져서 비정상 동작하게 되므로 가급적이면 서비스 관리자의 권한으로 적용하길 권장합니다.
- 획득한 권한은 90일 동안 유지되므로, 기한이 만료되면 access token을 재발급 받아야 합니다.

2. 준비 사항

- OAuth Client 등록 API 호출을 위해 HTTPS API를 호출할 수 있는 도구가 필요합니다.
 - [Google Chrome](#): Thing+ Portal에 로그인할 때 사용합니다.
 - [Postman](#): 원하는 HTTPS API를 호출할 때 사용하는 Google Chrome App입니다.
 - [Postman Interceptor](#): Thing+ Portal에 로그인했을 때 생성된 쿠키를 Postman에서 공유할 수 있도록 지원하는 Google Chrome Extension입니다.
 - 위의 도구를 사용하지 않더라도 Thing+ Portal에서 로그인한 상태로 HTTPS POST API를 호출할 수 있는 도구를 사용하면 수행 가능합니다.
 - [Thing+ Support 사이트](#)를 참고하세요.
 - 사용된 API에 대해 좀더 자세히 알고 싶을 경우 [Thing+ API Reference](#)를 참고하세요.

3. OAuth Client 생성

1. Chrome 에서 Thing+ Portal에 접속하여 서비스 관리자 계정으로 로그인합니다.
2. Postman 을 실행하고, Postman Interceptor 를 on 한 다음 아래 API를 호출합니다.

- 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 client의 이름으로 자유롭게 입력하시면 됩니다.
 - reqId: auth client의 ID로 access token을 얻을 때 입력하게 됩니다. 정하신 값을 입력하시면 됩니다.
 - clientSecret: auth client의 secret 값으로 access token을 얻을 때 입력하게 됩니다. 정하신 값을 입력하시면 됩니다.
 - scopes: auth client에 부여할 권한을 나열합니다. scopes에 들어갈 수 있는 값은 [link](#)를 참고하세요.
- 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 획득

1. Application에서 다음 API를 이용하여 access token을 획득합니다.

- 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: "password"라는 문자열 그대로 입력합니다. (사용자의 password를 입력하는 것이 아닙니다. 주의하세요.)
- client_id: /v2/authClient API로 생성한 auth client의 ID값을 입력합니다.
- clientSecret: /v2/authClient API로 생성한 auth client의 secret값을 입력합니다.
- username: auth client 생성 시 Thing+ Portal에 로그인했던 사용자(일반적으로 서비스 관리자) ID를 입력합니다.
- password: auth client 생성 시 Thing+ Portal에 로그인했던 사용자 비밀번호의 md5 hash 값을 입력

합니다.

- OSX나 Linux에서는 아래의 명령어를 이용하여 md5 hash 값을 구합니다.

- OSX

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

- Linux

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

- 위의 명령이 동작하지 않는 환경에서는 인터넷 상의 md5 hash generator 등을 이용하여도 비밀번호의 md5 hash 값을 구할 수도 있지만, 보안에 유의하시기 바랍니다.
- [JavaScript-MD5](#)를 이용할 수도 있습니다.

- Response

- Example

```
{  
  "access_token": "2yJ0eXAIoiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiIxIiwia2xpZW50SWQiOiJ0ZXN0Q2xpZW50SWQiLCJpYXQiOiJlOTU0MzYwMzksImV4cCI6MTUwMzIxMjAzOX0.dQ65zRCgRml96fTc8CDnExAukrFPSLd7NzDlUkf4eYk",  
  "token_type": "Bearer"  
}
```

2. 획득한 access token는 API를 호출할 때 아래와 같은 방식으로 사용합니다.

- Header에 `Authorization` 필드를 추가하고 value는 `token_type` 과 `access_token` 을 1칸 띄워 씁니다.
- curl 예제

```
$ curl -H "Authorization: Bearer 2yJ0eXAIoiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiIxIiwia2xpZW50SWQiOiJ0ZXN0Q2xpZW50SWQiLCJpYXQiOiJlOTU0MzYwMzksImV4cCI6MTUwMzIxMjAzOX0.dQ65zRCgRml96fTc8CDnExAukrFPSLd7NzDlUkf4eYk" https://api.sandbox.thingplus.net/v2/gateways
```

5. Gateway 등록

1. Site ID 구하기

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

- Example

```
{
  "statusCode": 200,
  "message": "OK",
  "data": [
    {
      "billing": "site:1",
      "code": "iotservice",
      "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"
    }
  ]
}
```

- response의 `data` 에 site의 목록이 반환됩니다. (일반적으로는 1개이지만, 여러 개의 site를 생성했을 경우에는 여러 개의 site data가 반환됩니다.)
- 게이트웨이를 등록하고자 하는 site의 ID를 게이트웨이 등록 API에 적용합니다. (예제에서는 `"id": "1"` 이 site ID에 해당하는 필드입니다.)

2. Gateway Model ID 구하기

- 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"
    }
}
}
}
}
}

```

- 게이트웨이 모델은 등록할 게이트웨이의 특성을 정의한 것입니다. 어떤 형식의 게이트웨이 ID를 사용할지, 어떤 디바이스와 센서를 등록할 수 있는지 등에 관한 정보가 들어 있습니다.
- 이 문서에서 설명하는 방식을 이용하여 게이트웨이를 등록하기 위해서는 게이트웨이 모델 ID `52` (Open API Gateway)를 사용합니다. 만약 다른 게이트웨이 모델을 사용할 경우, 본 API를 호출할 때 URL에서 `52` 대신 다른 ID를 대입해서 사용할 수 있습니다.

3. Gateway ID 생성하기

- `Open API Gateway` 의 경우 UUID를 게이트웨이 ID로 사용합니다.
- 새로 등록할 게이트웨이의 ID에 사용할 UUID를 구합니다.
 - OSX의 경우 `uuidgen` 명령으로 생성할 수 있습니다.
 - Linux(Ubuntu)의 경우 `uuid` 라는 package 설치 후 `uuid` 명령으로 생성할 수 있습니다.
 - UUID는 이 방법 외에도 다양한 방법으로 구할 수 있습니다.
- 위에서 구한 UUID에서 `-` 를 제거하고, 대문자가 있을 경우 소문자로 변경하여 게이트웨이 ID로 사용합니다.

```

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

```

4. Gateway 등록하기

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

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

- reqId: UUID로 생성한 ID를 사용합니다.
- siteId: 위에서 구한 사이트 ID를 사용합니다.
- model: 위에서 구한 게이트웨이 모델 ID를 사용합니다.
- name: 게이트웨이의 이름으로 자유롭게 입력하시면 됩니다.
- 명시한 항목 외의 옵션에 대해서는 [Thing+ API Reference](#)를 참고하세요.

◦ 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: 전송 주기 로서 게이트웨이가 센서에서 측정한 값을 Thing+로 전달하는 주기(msec)를 의미합니다. 이 값은 Thing+ Portal의 게이트웨이 관리 에서 수정할 수 있습니다.

6. Device 등록

1. Device 등록하기

- URL: `https://api.sandbox.thingplus.net/v2/gateways/{owner}/devices`
 - {owner}: 디바이스가 연결된 게이트웨이의 ID를 입력합니다. 위에서 등록한 게이트웨이 ID를 사용합니다.
- Method: POST
- Body:

- Example

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

- reqId: UUID를 이용하여 디바이스 ID를 생성하여 사용합니다.
- name: 디바이스의 이름으로 자유롭게 입력하시면 됩니다.
- model: 게이트웨이 모델을 조회했을 때 `deviceModels` 에 있는 디바이스 모델 ID를 사용합니다. 이 문서에서 설명하는 방식을 이용하여 디바이스를 등록하기 위해서는 디바이스 모델 ID `open-api-device-v1.0` 를 사용합니다. 만약 다른 디바이스 모델이 게이트웨이 모델에 등록되어 있을 경우, 본 API를 호출할 때 다른 디바이스 모델 ID를 대입해서 사용할 수 있습니다.
- 명시한 항목 외의 옵션에 대해서는 [Thing+ API Reference](#)를 참고하세요.

- 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 등록

1. Sensor 등록하기

- URL: `https://api.sandbox.thingplus.net/v2/gateways/{owner}/sensors`
 - `{owner}`: 센서가 연결된 게이트웨이의 ID를 입력합니다. 위에서 등록한 게이트웨이 ID를 사용합니다.
- Method: POST
- Body:

- Example

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

- `reqId`: UUID를 이용하여 센서 ID를 생성하여 사용합니다.
 - `name`: 센서의 이름으로 자유롭게 입력하시면 됩니다.
 - `type`: 게이트웨이 모델을 조회했을 때 `deviceModels.sensors` 에서 해당하는 센서 타입 값 (`type`)을 찾아 사용합니다.
 - `driverName`: 게이트웨이 모델을 조회했을 때 `deviceModels.sensors` 에서 `type` 에 해당하는 `driverName` 값을 사용합니다.
 - `model`: 게이트웨이 모델을 조회했을 때 `deviceModels.sensors` 에서 `type` 에 해당하는 `model` 값을 사용합니다.
 - `category`: `sensor` 라는 문자열 그대로 입력합니다. 현재 HTTPS 방식으로 `actuator` 는 지원하지 않습니다.
 - `deviceId`: 센서가 연결된 디바이스 ID를 입력합니다.
 - 명시한 항목 외의 옵션에 대해서는 [Thing+ API Reference](#)를 참고하세요.
- Response
 - Example

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

8. Status 전송

1. Gateway Status 전송하기

- URL: `https://api.sandbox.thingplus.net/v2/gateways/{id}/status`
 - `{id}`: 게이트웨이의 ID를 입력합니다. 위에서 등록한 게이트웨이 ID를 사용합니다.
- Method: PUT
- Body:
 - Example

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

- `validDuration`: `status` 가 유효한 기간(sec)을 의미합니다. 일반적으로 `reportInterval` 의 1.5배를 사용합니다.
- `value`: 게이트웨이의 상태입니다. 'on', 'off', 'err' 중의 하나를 보내야 합니다.
- 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**: **status** 가 언제까지 유효한지 나타내며, msec 단위의 Unix time입니다.
- 게이트웨이 상태는 **reportInterval** 마다 한 번씩 전송해야 합니다.
- 게이트웨이의 상태가 **on** 이라 하더라도 **expireAt** 의 시간이 지난 이후에는 해당 게이트웨이가 어떤 상태 인지 알 수 없습니다.

2. Sensor Status 전송하기

- URL: <https://api.sandbox.thingplus.net/v2/gateways/{owner}/sensors/{id}/status>
 - {owner}: 센서가 연결된 게이트웨이의 ID를 입력합니다. 위에서 등록한 게이트웨이 ID를 사용합니다.
 - {id}: 센서 ID를 입력합니다. 위에서 등록한 센서 ID를 사용합니다.

- Method: PUT
- Body:

▪ Example

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

- **validDuration**: **status** 가 유효한 기간(sec)을 의미합니다. 일반적으로 **reportInterval** 의 1.5배를 사용합니다.
- **value**: 센서의 상태입니다. 'on', 'off', 'err' 중의 하나를 보내야 합니다.

- 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: `status` 가 언제까지 유효한지 나타내며, msec 단위의 Unix time입니다.

- 센서 상태는 `reportInterval` 마다 한 번씩 전송해야 합니다.

- 센서의 상태가 `on` 이라 하더라도 `expireAt` 의 시간이 지난 이후에는 해당 센서가 어떤 상태인지 알 수 없습니다.

9. Sensor 값 전송

1. Sensor Value 전송하기

- URL: `https://api.sandbox.thingplus.net/v2/gateways/{owner}/sensors/{id}/series`
 - {owner}: 센서가 연결된 게이트웨이의 ID를 입력합니다. 위에서 등록한 게이트웨이 ID를 사용합니다.
 - {id}: 센서 ID를 입력합니다. 위에서 등록한 센서 ID를 사용합니다.

- Method: PUT

- Body:

- Example

- 1개의 센서값을 전송할 경우

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

- 여러 개의 센서값을 전송할 경우

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

- value: 센서값입니다. 문자열이나 오브젝트값을 넣어야 하며, 센서 타입에 맞아야 합니다. 단, 센서값이 숫자일 경우 문자열로 보냅니다.
- time: 센서값이 측정된 시간이며, msec 단위의 Unix time입니다.
- 여러 개의 센서값을 보낼 경우 배열로 묶어서 보낼 수 있습니다.

- Response

- Example

- 1개의 센서값을 전송했을 경우

```
{
  "statusCode": 200,
  "message": "OK",
  "data": {
    "name": "온도센서",
    "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"
  }
}
```

- 여러 개의 센서값을 전송했을 경우

```
{
  "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"
  }
}
```