



ThingPlug, 더 쉽게 시작하기

INDEX

01 ThingPlug 개요

- SKT ThingPlug
- oneM2M
- ThingPlug의 Resource 구조

02 ThingPlug API 기초

- Polling vs Subscription
- Case Study

03 ThingPlug App 예제

- ThingPlug API 익히기
- Python Clinet 사용하기



THINGPLUG

- ✓ SKT ThingPlug
- ✓ oneM2M
- ✓ ThingPlug의 Resource 구조



SKT ThingPlug

» ThingPlug

<https://thingplug.sktiot.com/>



디바이스

디바이스를 등록하시면,
API 및 앱을 통해 서비스가 가능합니다.

앱

다양한 IoT 앱을 개발하고, 그 가치를
창출할 수 있습니다.

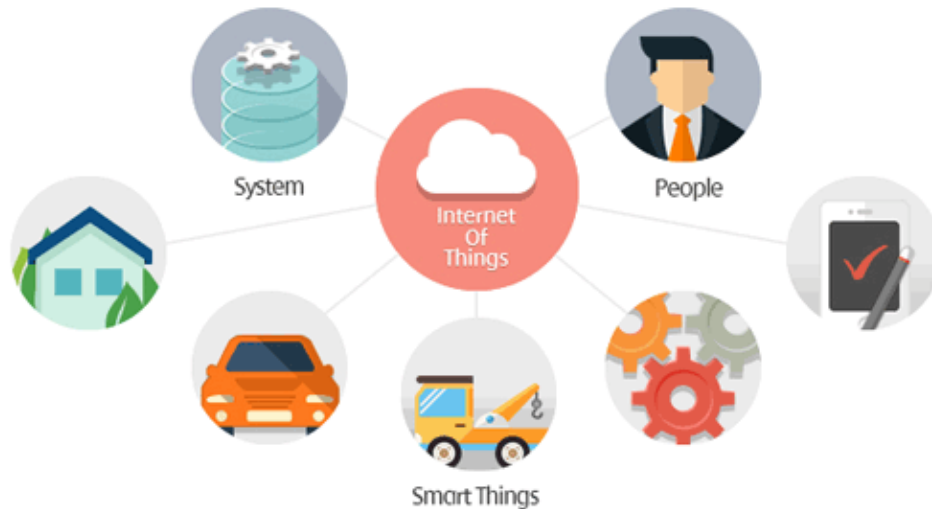
개발자 커뮤니티

개발관련 문의사항이나 해결방안을
공유할 수 있습니다.

통계

통계 자료를 통해 IoT 현황을 분석/판단
할 수 있습니다.

>> Internet Of Things

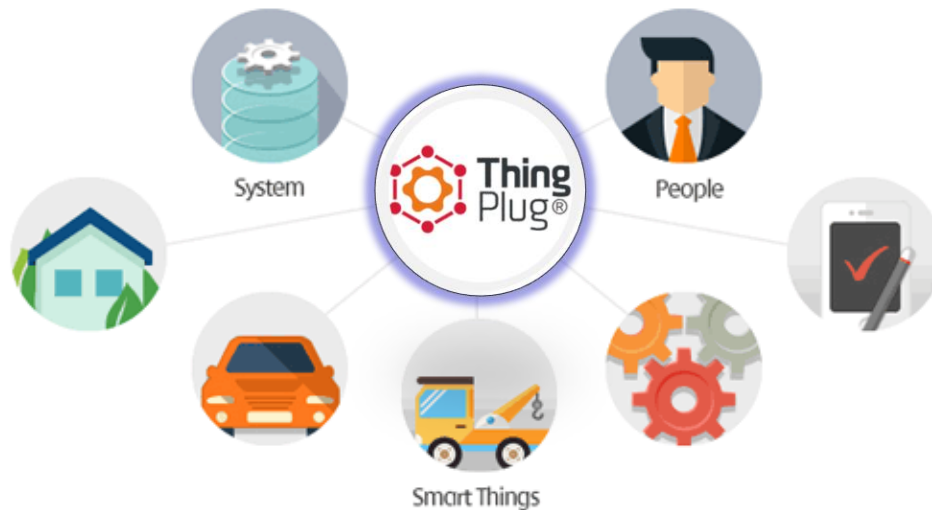


“ 가전, 모바일, 웨어러블 등의
각종 사물이 인터넷으로 연결.

데이터를 스스로 분석하고
사용자에게 제공하거나,
사물을 원격 조정.

”

>> IoT with ThingPlug

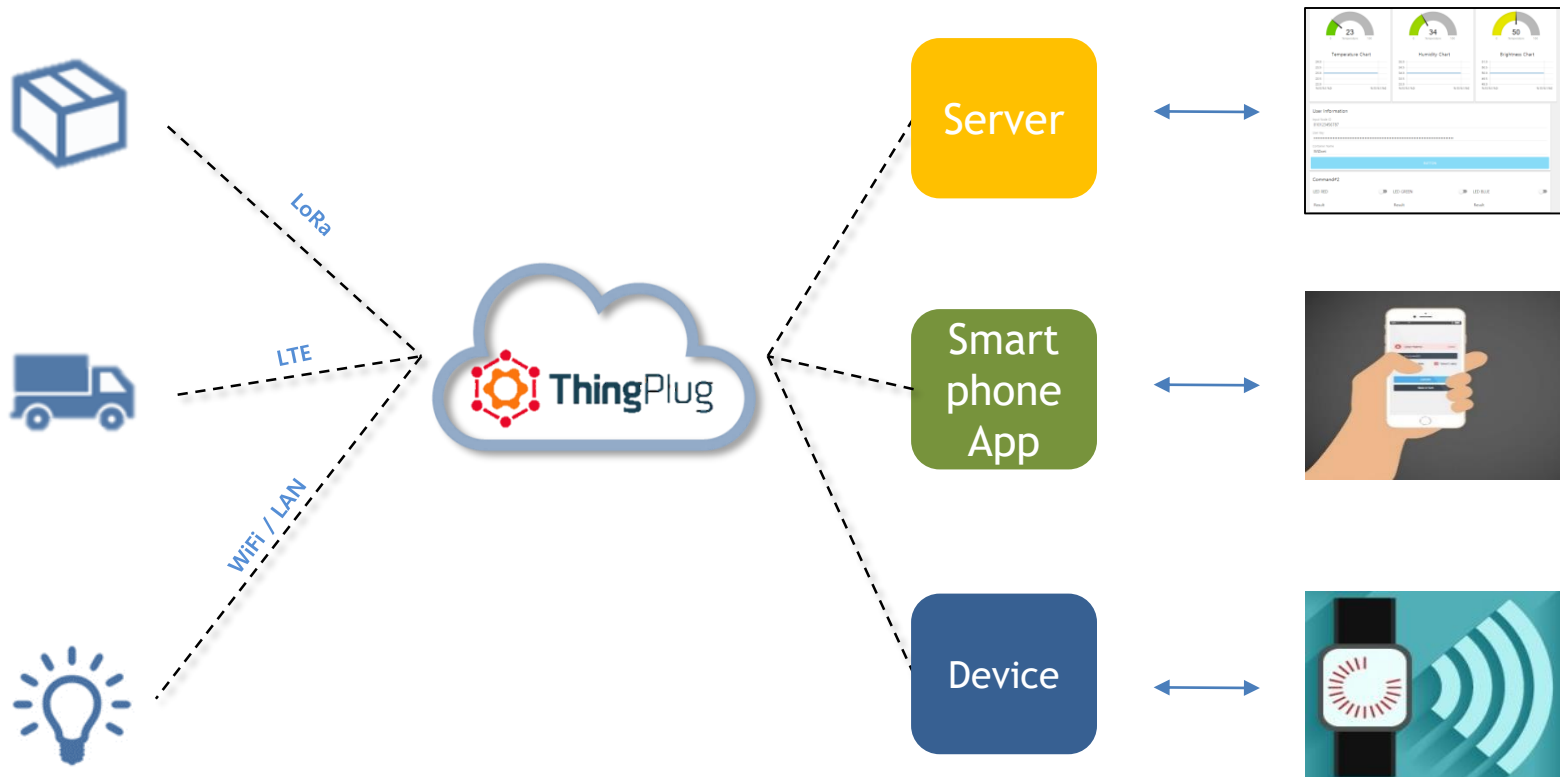


“ 개방형 IoT 플랫폼으로
IoT 서비스를 개발하기 위한
기반을 제공한다.

ThingPlug로 사물의 **연결** 및
관리, 서비스 개발, 데이터 **분석**
을 쉽게 할 수 있다.

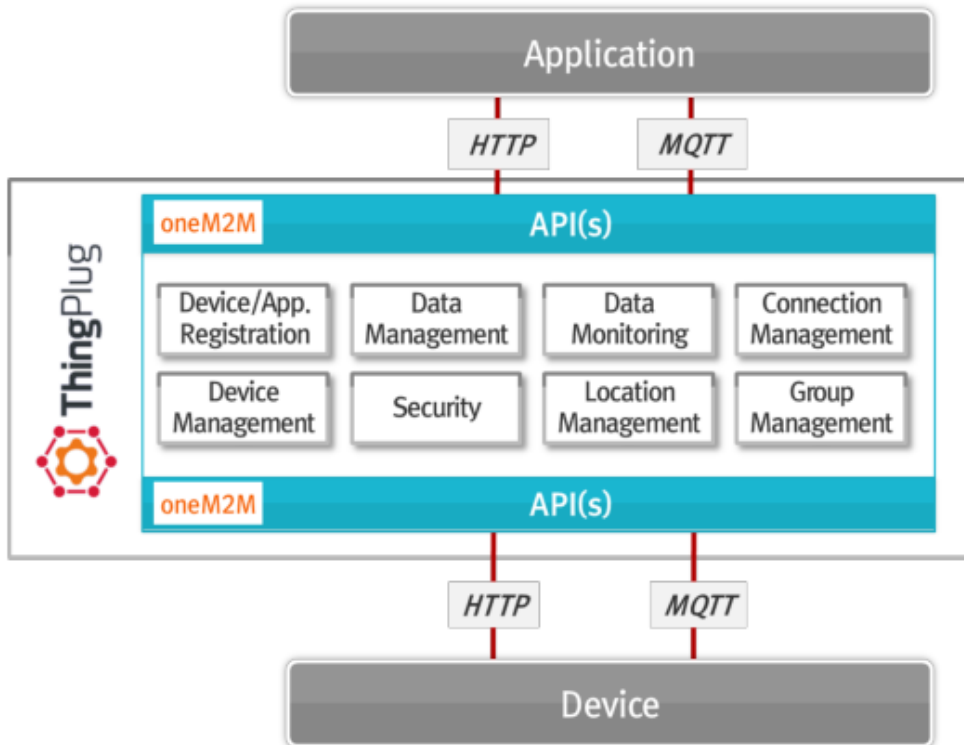
”

» IoT Device to Your Application





>> oneM2M



“

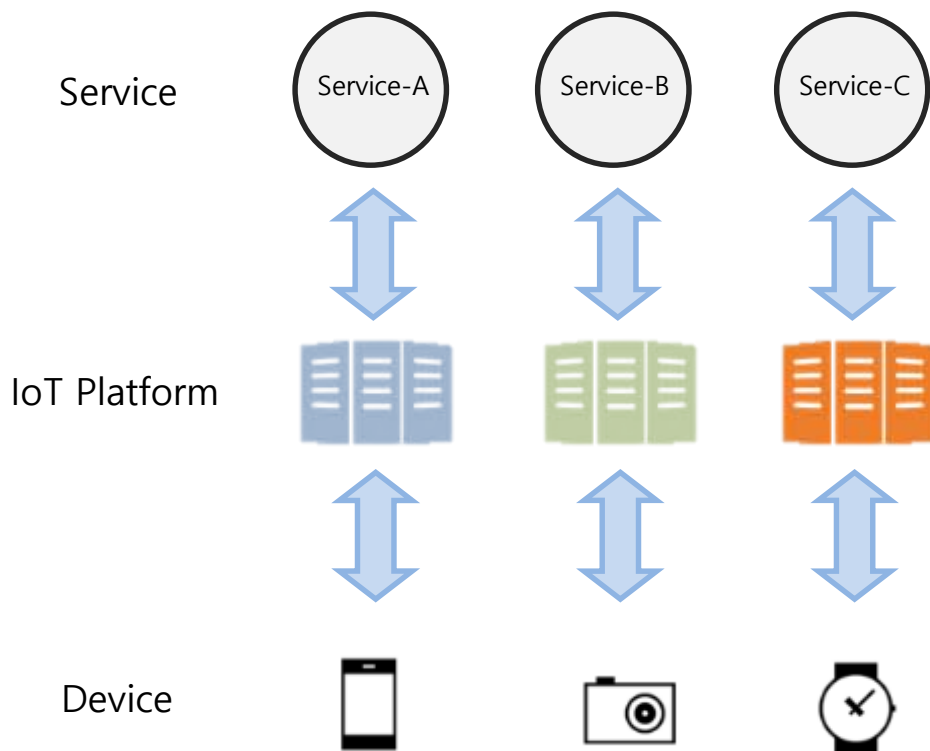
OneM2M 기반

ThingPlug는
IoT 어플리케이션 및
IoT 디바이스 개발자들이
공통적으로 필요로 하는
기능을 **API 형태**로 제공,

이에 접근하기 위해
HTTP와 **MQTT**를 사용합니다.

”

>> 기존의 서비스 방식

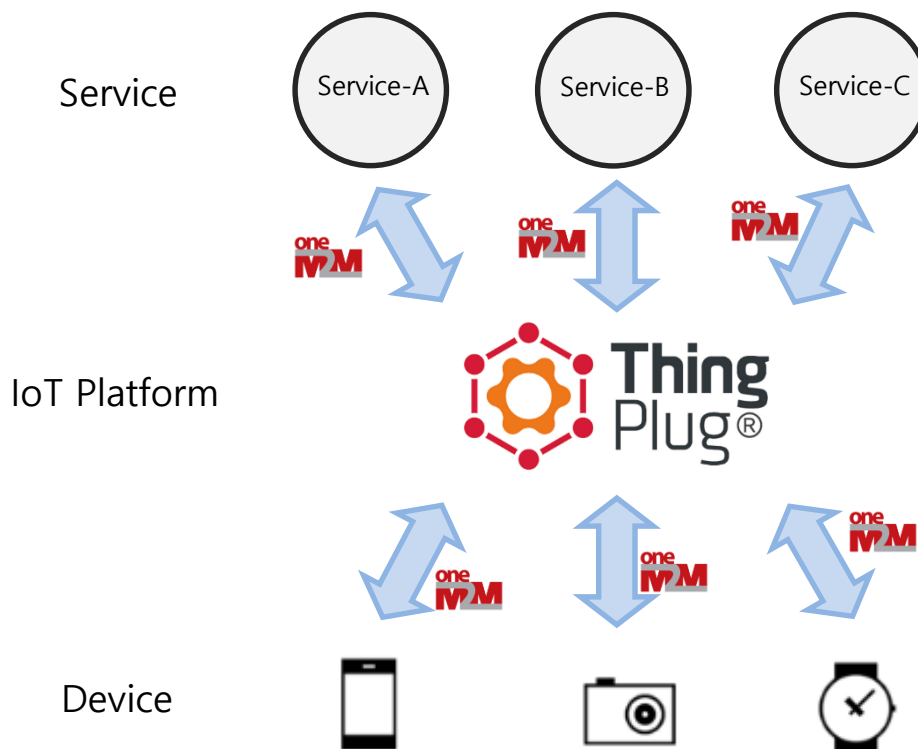


// 기존의 서비스는 제조사의 전용 디바이스, 전용 플랫폼, 전용 서비스를 하나의 솔루션으로 제공했다.

대부분의 IoT 서비스가 **비표준화** 상태로 제조사 자체 프로토콜을 사용해 개발이 진행되고 있다.

//

>> oneM2M



“ IoT, M2M 분야 표준화 단체
들이 모여 IoT 서비스에 필요
한 공통의 요구사항을 바탕으
로 **oneM2M이라는 표준 프로
토콜을** 개발했다.

기존 서비스에 다른 디바이스나
App을 **연동하기 쉬워져** 서비스
확장이 쉬워질 수 있다.

우리나라에는 TTA, SK텔레콤
등이 개발에 적극 참여했다.

”

>> oneM2M

SKT IoT 플랫폼 '씽플러그' oneM2M 표준 인증 획득

국제 표준 준수 검증 완료, 세계 시장 진출 준비 마쳐

조양준 기자 | 2016-05-10 10:40:02 | IT과학



SK텔레콤은 이 회사의 사물인터넷(IoT) 플랫폼인 '씽플러그(ThingPlug)'가 한국정보통신기술협회(TTA) 'oneM2M 표준 인증'을 획득했다고 10일 밝혔다.

oneM2M 표준 인증은 국내 IoT 플랫폼이 국제 표준인 'oneM2M'을 준수하고 있는지 검증하는 프로그램이다. TTA는 지난해부터 국내 IoT 제품·서비스가 글로벌 경쟁력을 확보할 수 있도록 서버플랫폼, 센서 등에 대한 인증 기준을 마련했으며, 지난 4월 국내 IoT 플랫폼을 대상으로 검증에 들어갔다.

LGU+ 'oneM2M 기반 IoT 플랫폼' 호환성 국제 검증

김형영 기자 | 2016-05-16 15:41:59 | IT과학



LG유플러스는 자사의 oneM2M 기반 사물인터넷(IoT) 상용플랫폼이 상호호환성 국제행사에서 검증을 마쳤다고 16일 밝혔다.

oneM2M은 사물인터넷 시장에서 기기 간 공통의 의사소통 방식을 결정하는 중요한 열쇠이기 때문에 이번 검증이 중요한 의미를 가진다는 것이 LG유플러스의 설명이다.

oneM2M은 세계 IoT 표준 중 하나로, oneM2M 상호호환성 국제 행사는 한국정보통신기술협회(TTA)와 유럽전기통신표준협회(ETSI)가 공동 주관하며 올해는 한국에서 이달 10일부터 13일까지 진행했다. 이번 행사는 oneM2M 표준 기기 및 서비스 간 상호호환성 확보를 목표로 진행됐으며 전세계 28개 업체가 참가했다. LG유플러스는 이 자리에서 플랫폼의 실제 상용 사례에 기반한 아키텍처(구조) 구성, 구현 기술 등도 설명했다.

방송/통신 KT IoT 메이커스, oneM2M 릴리즈1 공식인증

세계 IT비즈니스 플랫폼을 글로벌로 진출 확대

박수형 기자 | 입력 : 2016.11.24.11:27 | 수정 : 2016.11.24.11:27

KT IoT 플랫폼 'IoT 메이커스(Makers)'가 23일 한국정보통신기술협회(TTA)에서 진행된 oneM2M Release 1 공식 인증 시험을 통과했다.

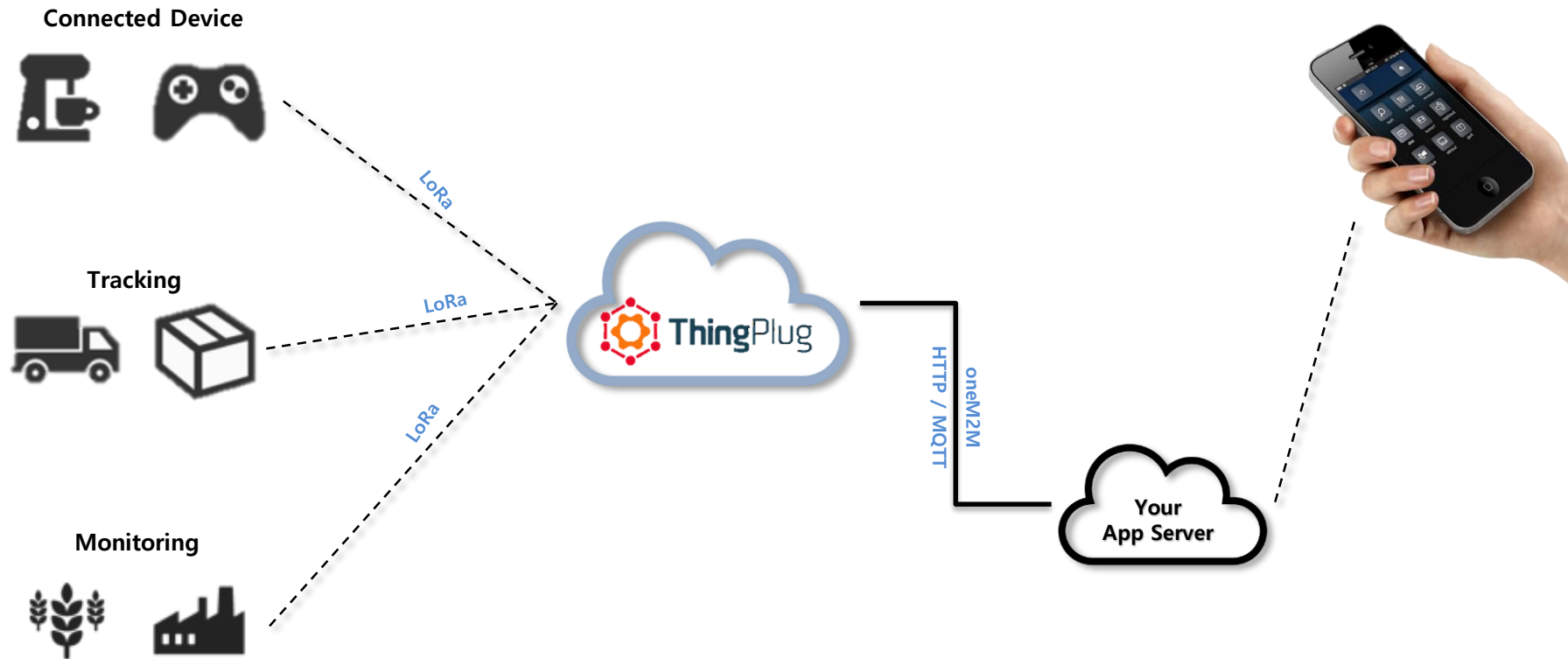
표준 인증은 oneM2M 국제 표준 단체에서 IoT플랫폼이 국제표준을 준수하고 있는지 검증하는 프로그램이다.

KT는 인증 획득에 따라 국내 통신사 중 유일하게 서버 플랫폼 및 디바이스 플랫폼에 대해 HTTP, CoAP, MQTT 등 프로토콜과 메시지 형식 XML, JSON을 모두 지원할 수 있게 됐다고 설명했다.

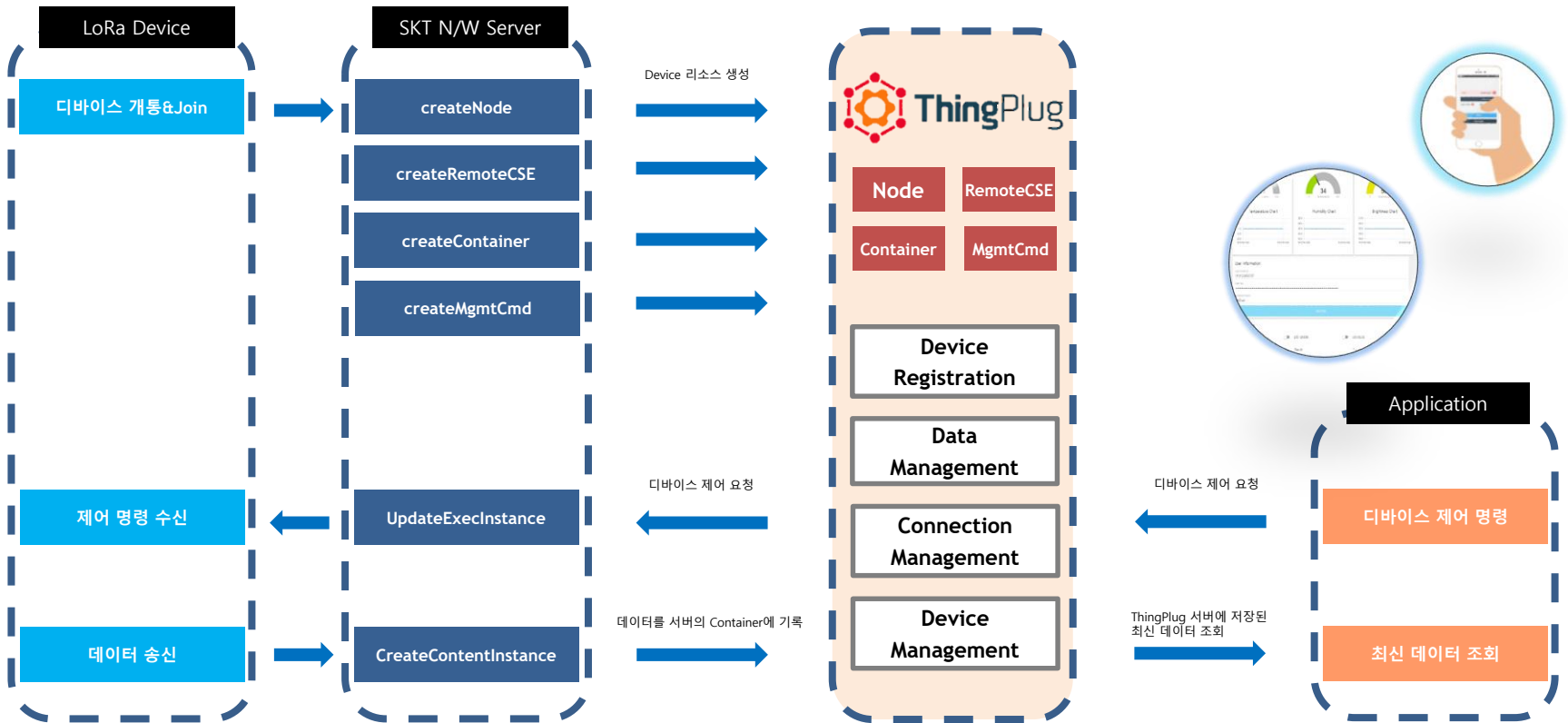
“
SKT ThingPlug 뿐만 아니라
국내 통신사, 중소형 IoT 플랫폼
들도 oneM2M 프로토콜을
기반으로 개발되고 있다.
”



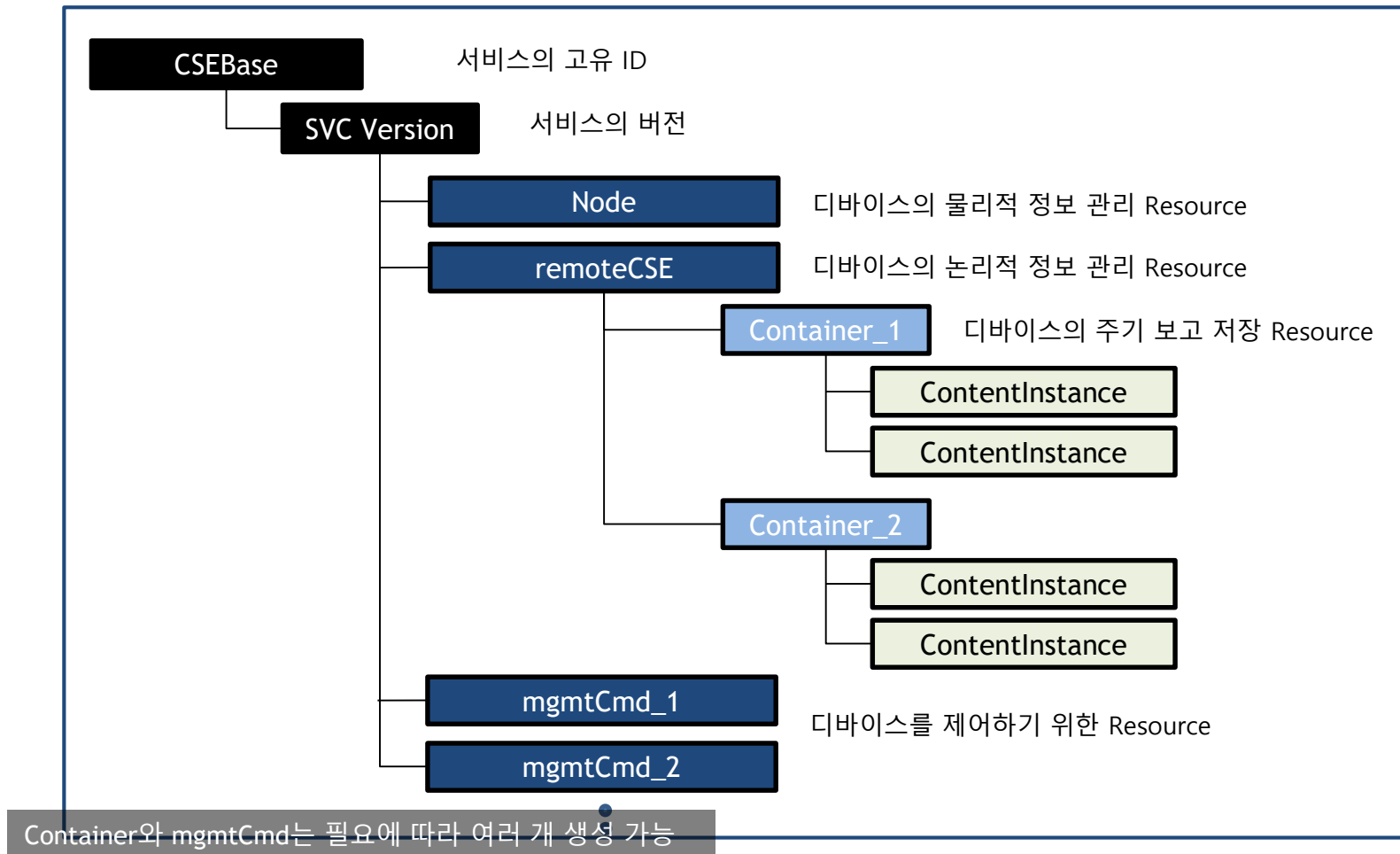
>> SKT LoRa to Your APP



>> SKT LoRa to Your APP



ThingPlug의 oneM2M Resource Structure.



>> ThingPlug의 oneM2M Resource Structure.

디바이스의 최신 데이터를 조회하기 위해서는 아래와 같은 Query를 사용한다.

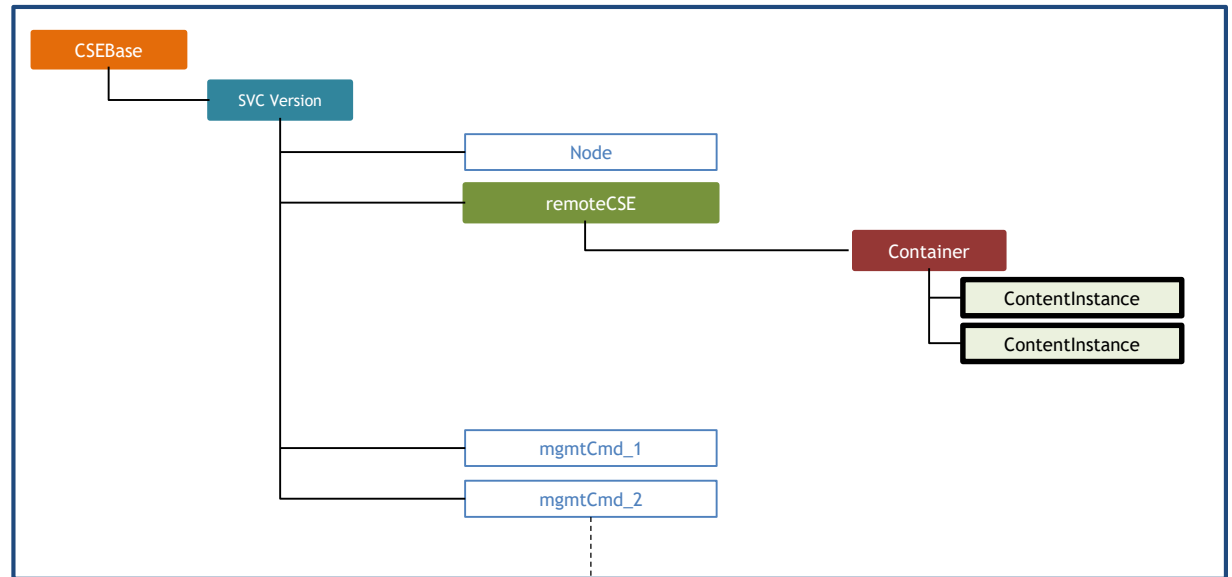
https://onem2m.sktiot.com:9443/0000000000000004/v1_0/remoteCSE-00000004dd2544ffff03777/container-LoRa/latest

CSEBase / Version / remoteCSE / Container

CSEBase = APP EUI

remoteCSE = LTID

Container = LoRa



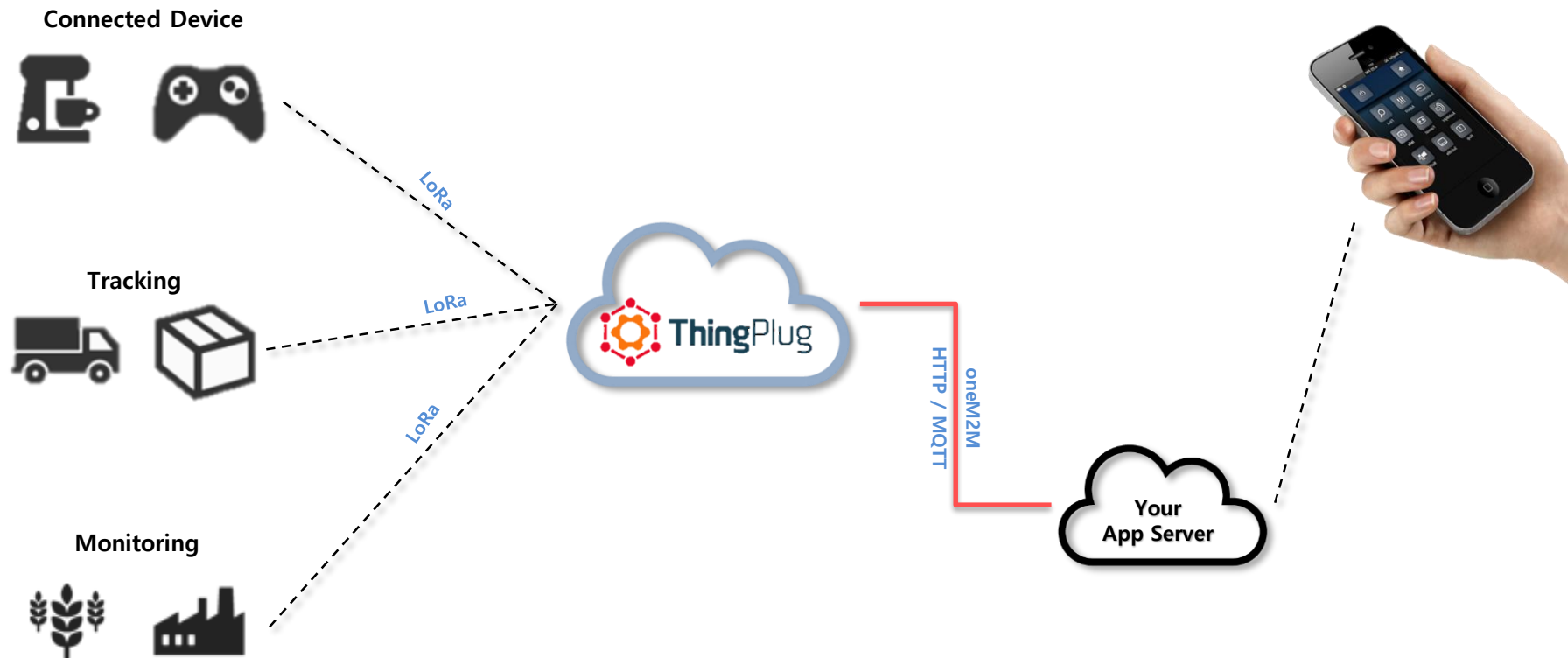
같은 방법으로 제어 결과도 조회할 수 있다.



THINGPLUG API

- ✓ Polling vs Subscription
- ✓ Case Study

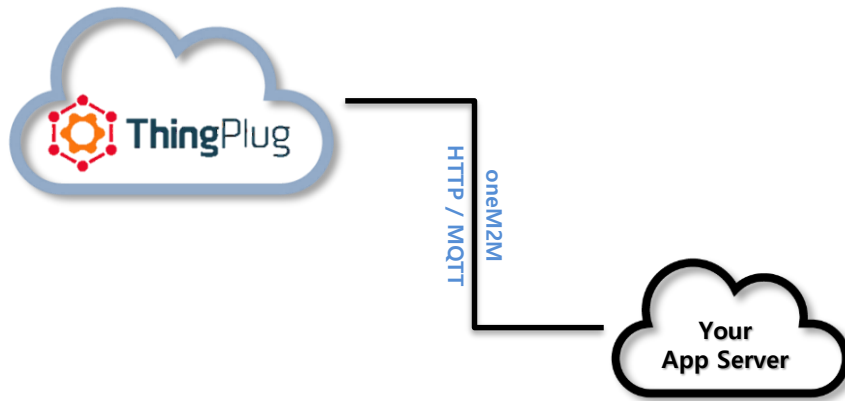
>> SKT LoRa to Your APP





Polling vs Subscription

>> Polling과 Subscription의 차이



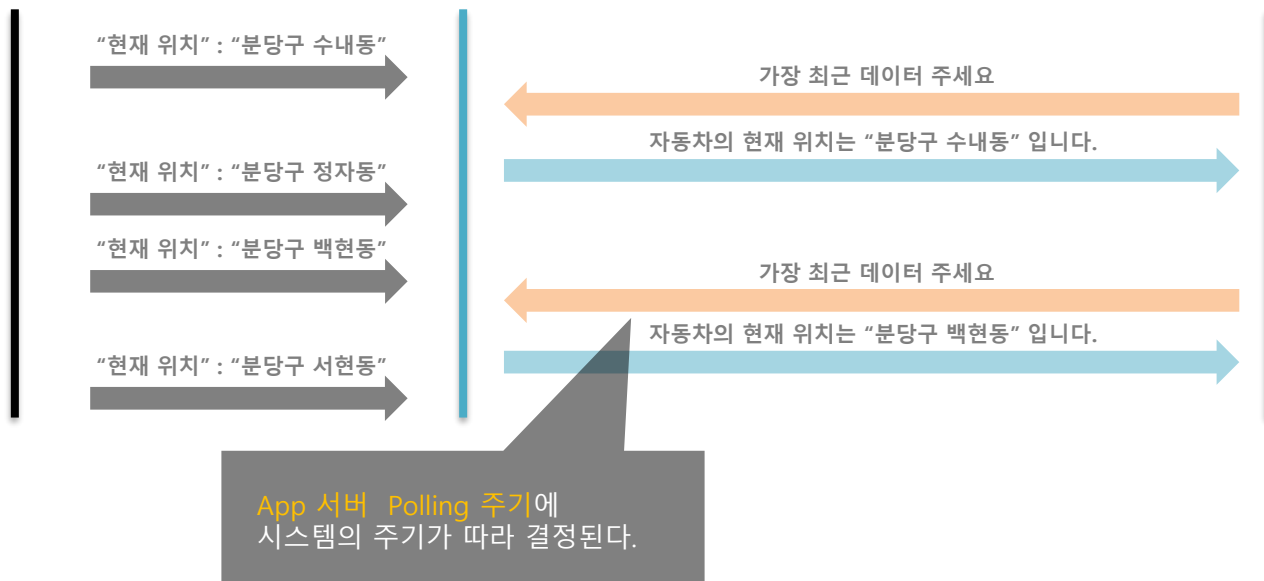
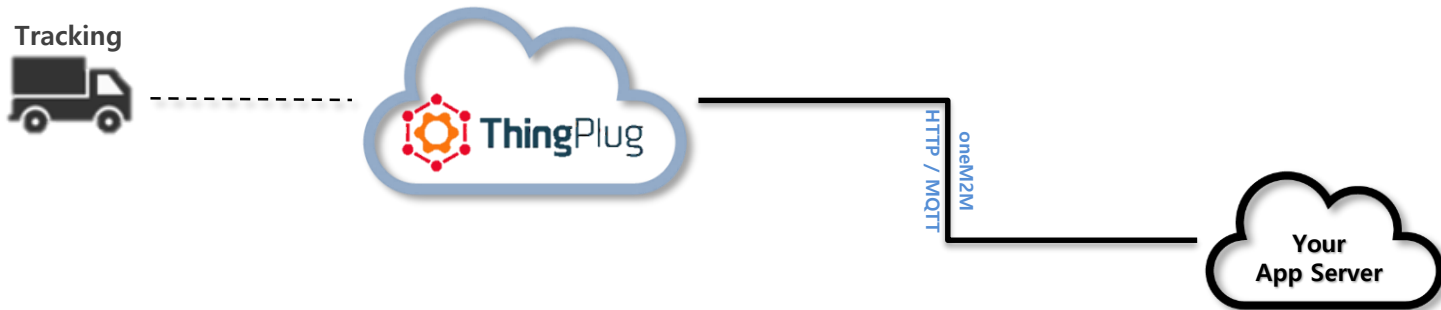
Polling (HTTP, MQTT)

- App 서버의 요청으로 데이터 조회
- App 서버에서 조회 주기 설정
- 디바이스 주기와 App 서버 polling 주기에 따라 데이터 유실 가능성이 있다

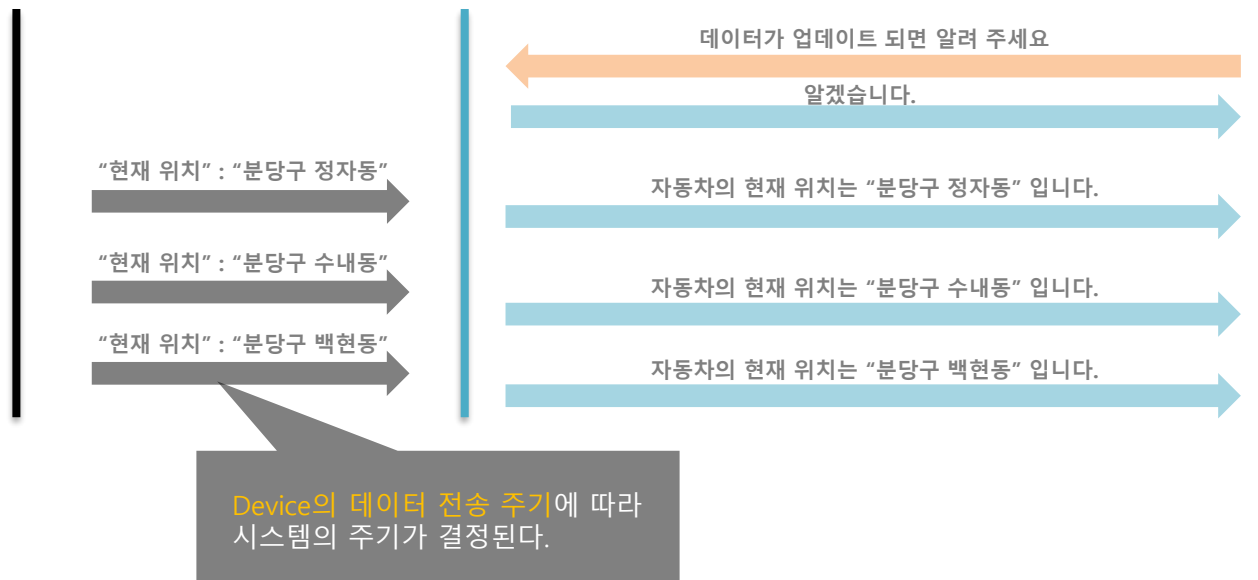
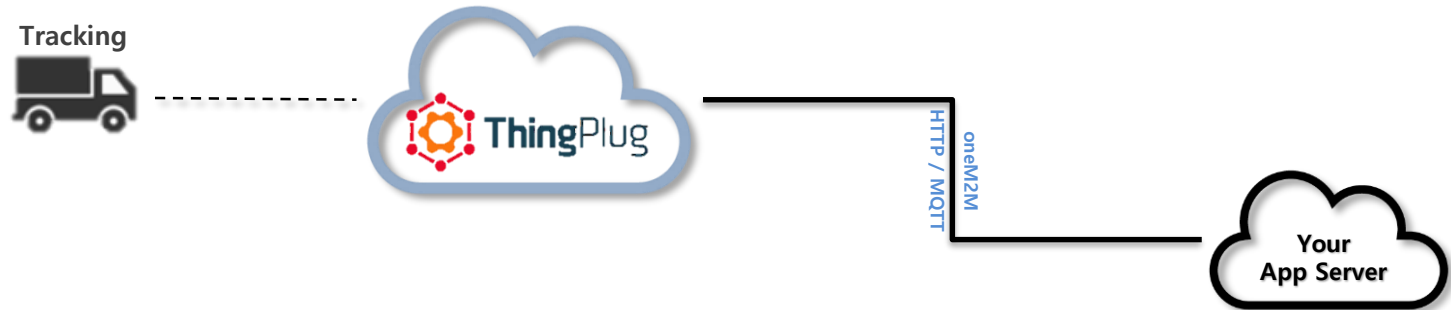
Subscription (HTTP, MQTT)

- 데이터가 생성되면 자동으로 Push
- Device 서버에서 조회 주기 설정
- 실시간으로 데이터를 App 서버에서 확인 가능

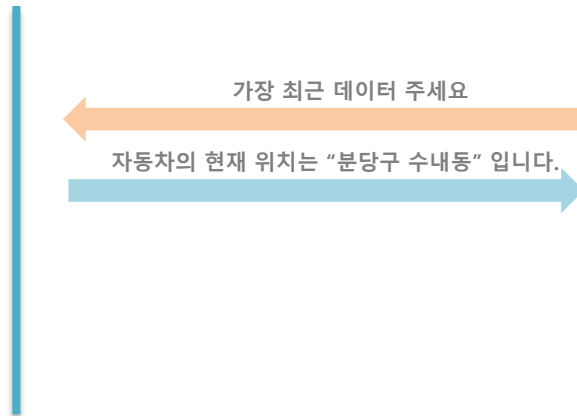
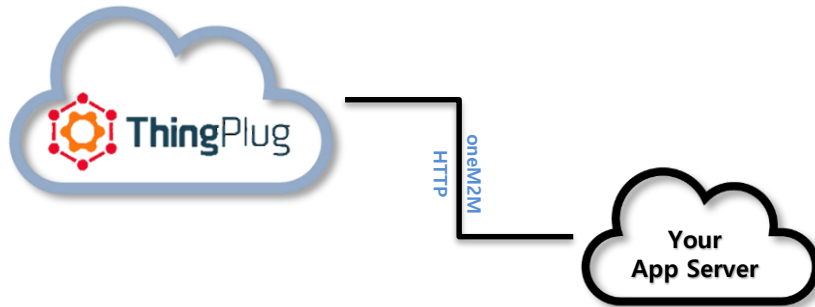
>> Polling 방식의 데이터 수집 방법



>> Subscription 방식의 데이터 수집 방법



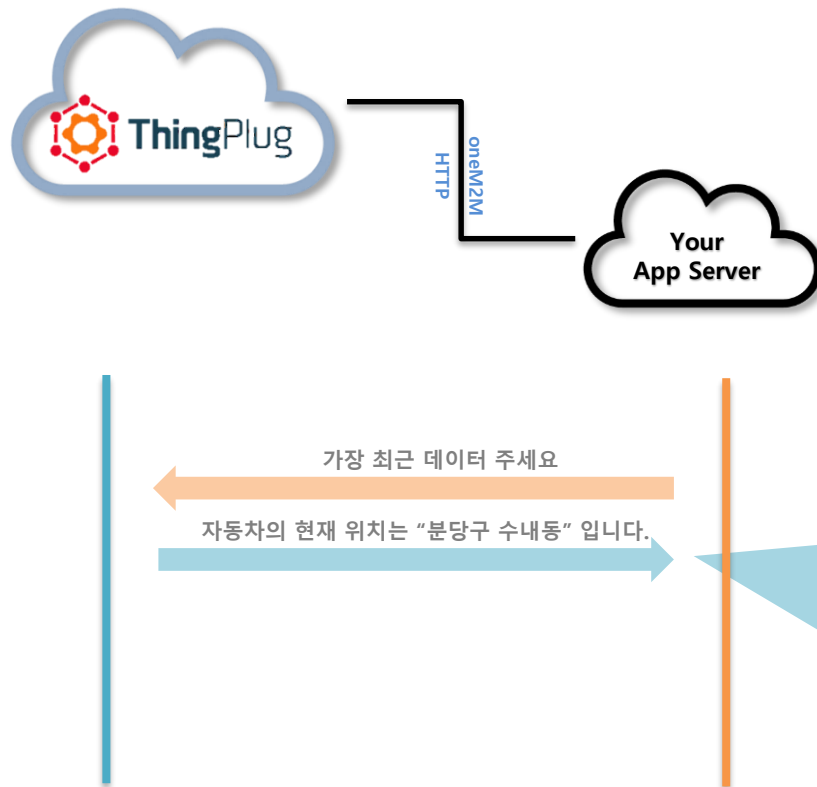
>> Polling 방식의 데이터 수집 방법 (REQ)



GET `/{{APP EUI}}/v1_0/remoteCSE-{{Dev ID}}/container-LoRa/latest` HTTP/1.1

Host: onem2m.sktiot.com:9443
X-M2M-RI: 0123456789_0002
X-M2M-Origin: {{Dev ID}}
ukey: {{User key}}

>> Polling 방식의 데이터 수집 방법 (RESP)



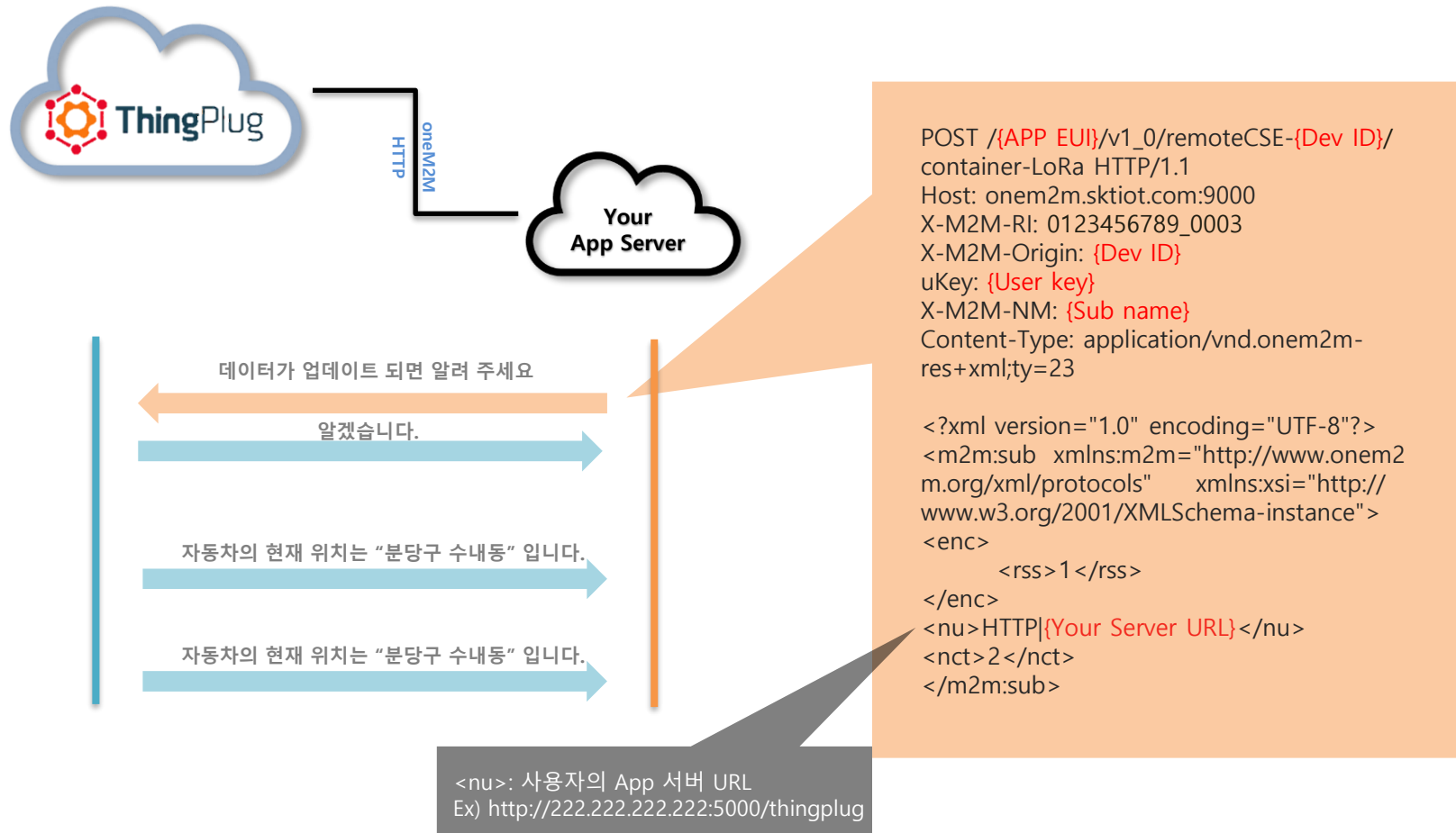
<ct>: 디바이스가 데이터를 생성한 시간
<con>: 디바이스가 보낸 데이터

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
<m2m:cin  
  xmlns:m2m="http://www.onem2m.org/xml/protocols"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

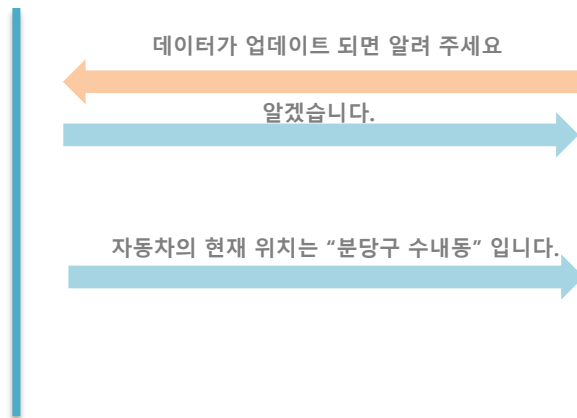
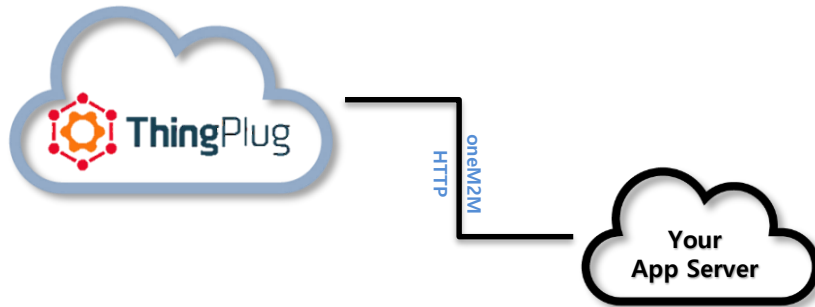
```
  <ty>4</ty>  
  <ri>CI000000000000070218874</ri>  
  <rn>CI000000000000070218874</rn>  
  <pi>CT000000000000000004847</pi>  
  <ct>2017-03-29T16:55:43+09:00</ct>  
  <lt>2017-03-29T16:55:35+09:00</lt>  
  <et>2017-03-30T16:55:43+09:00</et>  
  <st>193</st>  
  <cr>RC00000000000000000382125</cr>  
  <cnf>LoRa/Sensor</cnf>  
  <cs>12</cs>  
  <con>040101050101</con>
```

```
</m2m:cin>
```

>> Subscription 방식의 데이터 수집 방법 (REQ)



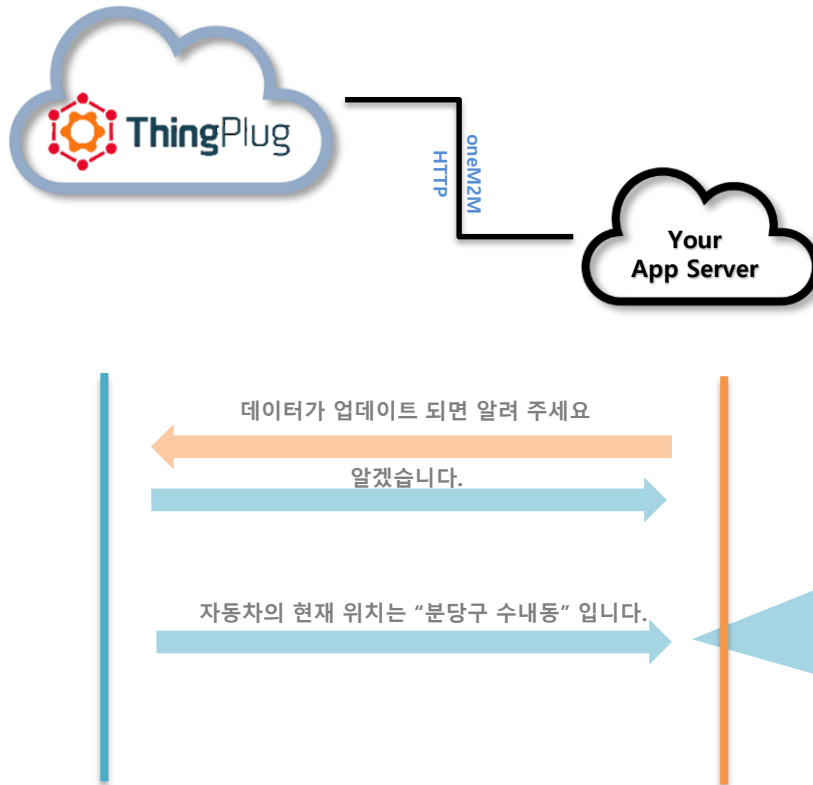
>> Subscription 방식의 데이터 수집 방법 (RESP)



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<m2m:sub
xmlns:m2m="http://www.onem2m.org/xml/protocols"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
  <ty>23</ty>
  <ri>SS0000000000000000025110</ri>
  <rn> {Sub name} </rn>
  <pi>CT0000000000000000013650</pi>
  <ct>2017-03-29T17:09:23+09:00</ct>
  <lt>2017-03-29T17:09:23+09:00</lt>
  <enc>
    <rss>1</rss>
  </enc>
  <nu>HTTP[{Your Server URL}]</nu>
  <nct>2</nct>
</m2m:sub>
```

>> Subscription 방식의 데이터 수집 방법 (PUSH)



```
POST /{URL} HTTP/1.1
X-M2M-Origin: {Dev ID}
X-M2M-RI: 27a7c56a-1074-4b98-8fdb-e20d6aed7839
Accept: application/xml
Content-Type: application/vnd.onem2m-ntfy+xml;charset=UTF-8
Host: {Your Server Host}
```

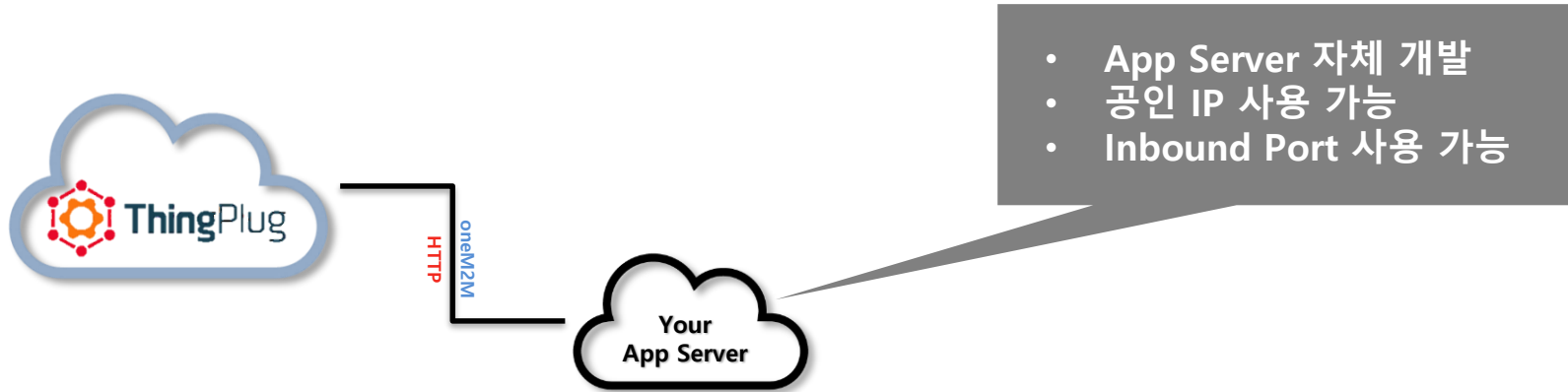
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<m2m:cin
xmlns:m2m="http://www.onem2m.org/xml/protocols"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<ty>4</ty>
<ri>CI000000000000046596250</ri>
<rn>CI000000000000046596250</rn>
<pi>CT0000000000000004353</pi>
<ct>2017-02-13T09:57:23+09:00</ct>
<lt>2017-02-13T09:57:23+09:00</lt>
<sr>/{APP EUI}/v1_0/remoteCSE-{Dev ID}/container-LoRa
/subscription-{Sub name}</sr>
<et>2017-03-15T09:57:23+09:00</et>
<st>8</st>
<cr>RC000000000000000381249</cr>
<cnf>LoRa/Sensor</cnf>
<cs>3</cs>
<con>040101050101</con>
</m2m:cin>
```



Case Study

>> Case 1, 일반적인 LoRa 서비스



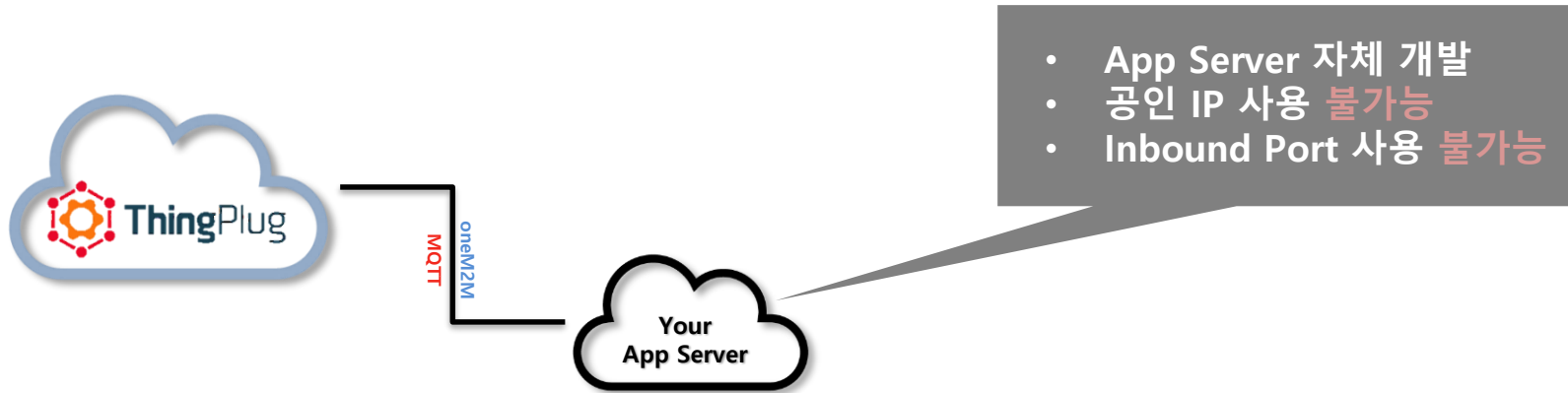
데이터 수집 방식

- Subscription HTTP

자체 App Server에 기능 구현

- 디바이스의 Subscription Binding 기능
- 데이터 수집, 분석, 처리를 구현

≫ Case 2, App Server Inbound 불가능한 상황



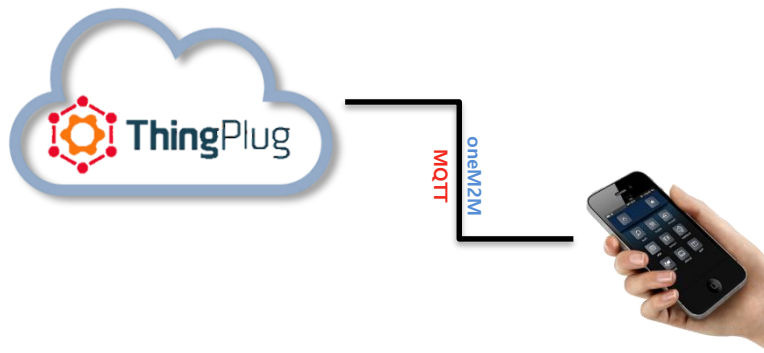
데이터 수집 방식

- Subscription **MQTT**

자체 App Server에 기능 구현

- 디바이스의 Subscription Binding 기능
- 데이터 수집, 분석, 처리를 구현

≫ Case 3, 스마트폰 App 등으로 직접 ThingPlug 연동



- 스마트폰 App 으로 구현

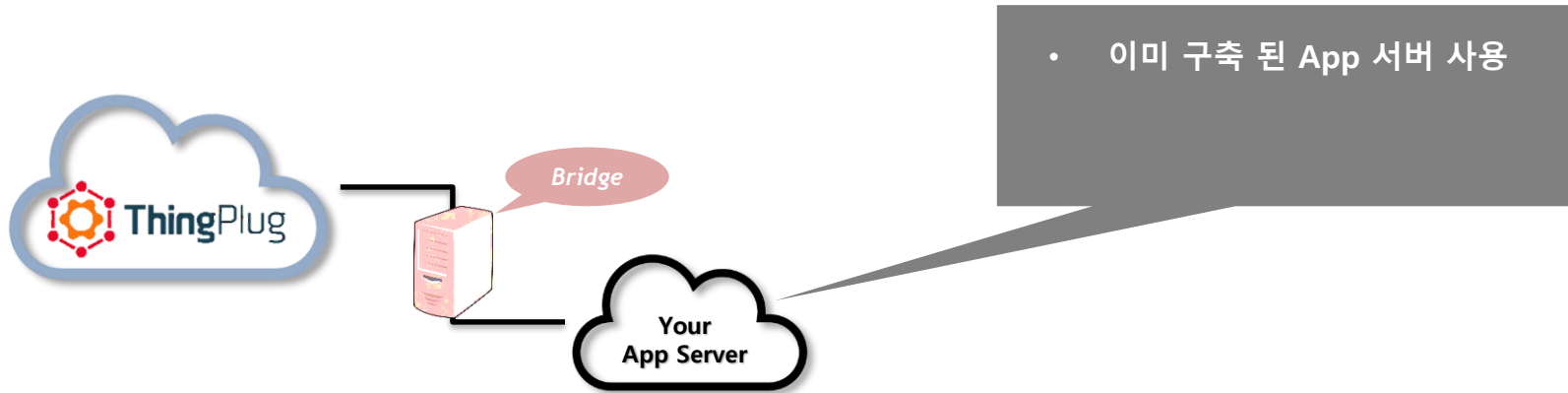
데이터 수집 방식

- Subscription *MQTT*

자체 스마트폰 App에 기능 구현

- 디바이스의 Subscription Binding 기능
- 데이터 수집, 분석, 처리를 구현

≫ Case 4, 기존 App Server를 변경 없이 사용하는 상황



Bridge의 데이터 수집 방식

- Subscription **HTTP & MQTT**

Bridge에 기능 구현

- 디바이스의 Subscription Binding 기능
- 데이터 수집, 분석, **기존 App Server로 전송 기능** 구현

>> Case 5, 연동 테스트 & 원하는 시점 데이터만 필요시



- App 서버가 원하는 시간에 데이터를 조회하고 싶은 경우
- 데이터 전송 주기가 긴 경우
- 연동 테스트

데이터 수집 방식

- Polling HTTP

기능 구현

- Polling으로 데이터 요청, 분석, 처리



THINGPLUG APP

- ✓ ThingPlug API 익히기
- ✓ Python Clinet 사용하기



ThingPlug API 익히기

» 사용자 인증키(Ukey) 확인

ThingPlug API를 사용하기 위해 필수로 사용되는 키, 마이페이지에서 확인

daniel 님 환영합니다. 로그아웃 정보수정 마이페이지

🏠 > 마이페이지 > 마이 IoT

마이 IoT

 **디바이스** 등록 2건

 **펌웨어 업그레이드** 신규 0건

 **매쉬업 API** 등록 0건

▶ 사용자 인증키

ZG9[REDACTED]SIRRA==

▶ 나의 디바이스

 **myDev2**
외식/유통, 미지정, 미지정
Alive
2017-02-28

 **myDev**
홈모니터링, 미지정, 미지정
Alive
2017-02-27

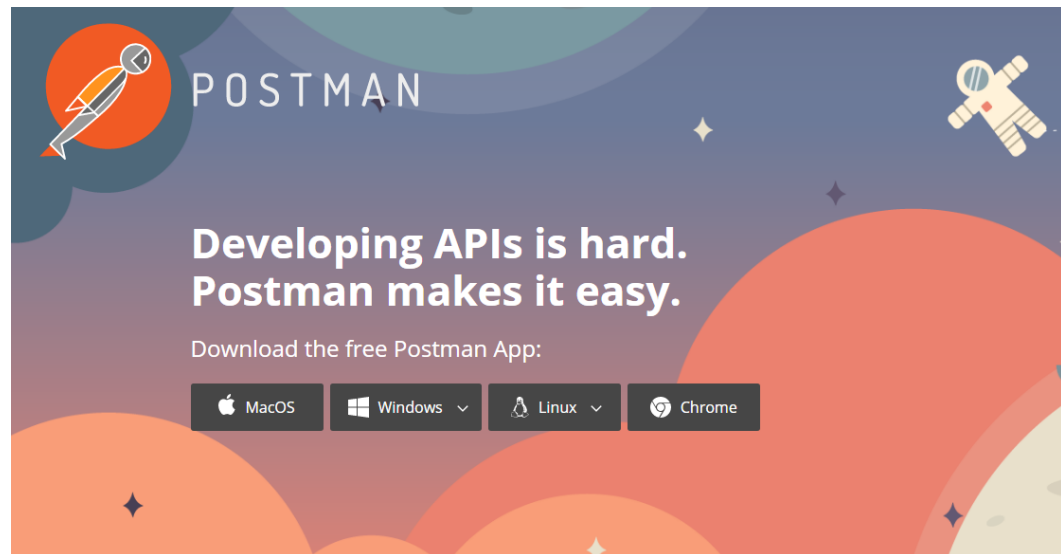
>> ThingPlug Application



➤➤ Postman 툴을 활용한 ThingPlug API 익히기



<https://www.getpostman.com/>



>> Postman 테스트 Script 만들기

강의자료\Utility\W01_Postman

이름	수정한 날짜	유형	크기
postman_src	2017-04-07 오후...	파일 폴더	
UserInfo.txt	2017-04-07 오후...	텍스트 문서	1KB
wiznet_postman.bat	2017-04-07 오후...	Windows 배치 파일	3KB



UserInfo.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

ThingPlugID daniel
ThingPlugKey ZG9qcUNscTA3
AppEUI 0000000000000000
DeviceID 00000004dd2544f...
DestServer http://209.88.180.134:8080



이름	수정한 날짜	유형	크기
debug	2017-04-07 오후...	파일 폴더	
postman_src	2017-04-07 오후...	파일 폴더	
ThingPlugAPI_daniel_00000004dd2544f...	2017-04-07 오후...	JSON 파일	22KB
UserInfo.txt	2017-04-07 오후...	텍스트 문서	1KB
wiznet_postman.bat	2017-04-07 오후...	Windows 배치 파일	3KB

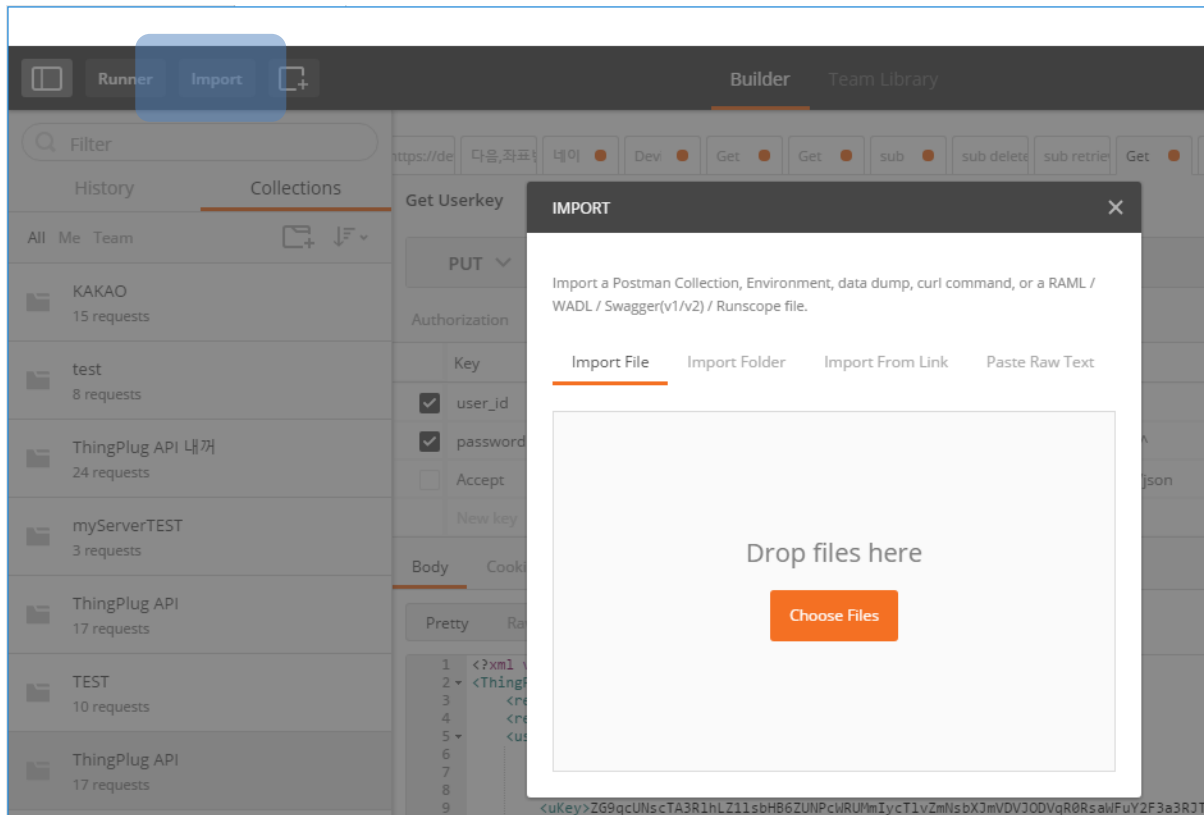
UserInfo.txt를 수정 후

wiznet_postman.bat 실행

ThingPlugAPI_xxx_xxx.json 생성

≫≫ Postman 테스트 Script를 import 하기

생성된 **ThingPlugAPI_xxx_xxx.json** 파일을 Postman에 **Import** 한다.



≫ User Key 확인

History Collections

All Me Team

ThingPlug(000000004dd2544fffff0...
9 requests

01. Login
PUT Get Userkey

02. Device List
GET Get Device List

03. App -> ThingPlug
GET Get latest Data
PUT Device Control
GET Get Control Result

04. Data Subscription
POST sub create (Data)
GET sub retrieve (Data)
PUT sub update (Data)
DEL sub delete (Data)

- 포털 ID와 PWD으로 User Key를 확인할 수 있다.
- User Key는 ThingPlug API를 사용하기 위해 필수로 사용되는 사용자 인증키이다.

▶ Get Userkey

PUT ▼ https://onem2m.sktiot.com:9443/ThingPlug?division=user&function=login Params

Authorization Headers (3) Body Pre-request Script Tests

<input checked="" type="checkbox"/>	user_id	{portal ID}	≡ ×
<input checked="" type="checkbox"/>	password	{portal Password}	≡ ×
<input type="checkbox"/>	Accept	application/json	≡ ×

>> User Key 확인

<Req>

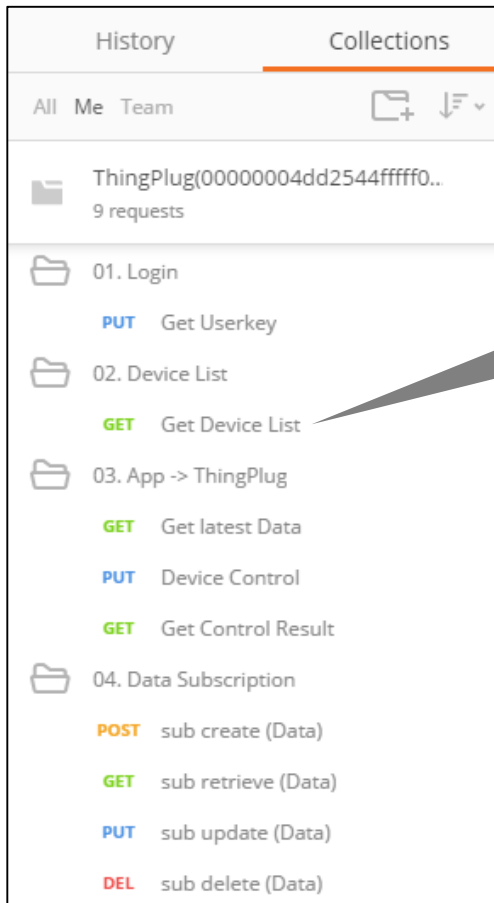
PUT /ThingPlug?division=user&function=login HTTP/1.1
Host: onem2m.sktiot.com:9443
user_id: <portal id>
password: <portal password>

<Resp>

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <ThingPlug>
3   <result_code>200</result_code>
4   <result_msg></result_msg>
5   <user>
6     <admin_yn>N</admin_yn>
7     </password></password>
8     <user_id>daniel</user_id>
9     <uKey>ZG9qcUNscTA3RlLnSStRRA==</uKey>
10  </user>
11 </ThingPlug>
```

ThingPlug API를 사용할 때
꼭 필요한 uKey

≫ 디바이스 리스트 확인



History Collections

All Me Team

ThingPlug(00000004dd2544ffff0...)
9 requests

01. Login
PUT Get Userkey

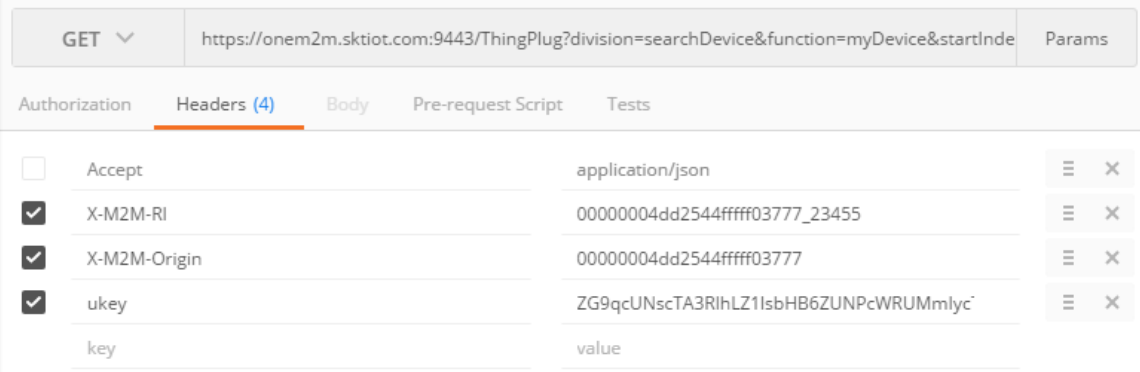
02. Device List
GET Get Device List

03. App -> ThingPlug
GET Get latest Data
PUT Device Control
GET Get Control Result

04. Data Subscription
POST sub create (Data)
GET sub retrieve (Data)
PUT sub update (Data)
DEL sub delete (Data)

- 사용자 계정에 등록된 디바이스 리스트를 얻을 때 사용한다.

▶ Get Device List



GET <https://onem2m.sktiot.com:9443/ThingPlug?division=searchDevice&function=myDevice&startInde> Params

Authorization Headers (4) Body Pre-request Script Tests

<input type="checkbox"/>	Accept	application/json	⋮ ×
<input checked="" type="checkbox"/>	X-M2M-RI	00000004dd2544ffff03777_23455	⋮ ×
<input checked="" type="checkbox"/>	X-M2M-Origin	00000004dd2544ffff03777	⋮ ×
<input checked="" type="checkbox"/>	ukey	ZG9qcUNscTA3RihLZ1lsbHB6ZUNPcWRUMmlyc'	⋮ ×
	key	value	

>> 디바이스 리스트 확인

startIndex: 조회할 디바이스의 Index
countPerPage: 조회할 디바이스 수

<Req>

GET /ThingPlug?division=searchDevice&function=myDevice&startIndex=1&countPerPage=1 HTTP/1.1
Host: onem2m.sktiot.com:9443
ukey: <user key>

<Resp>

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <ThingPlug>
3   <result_code>200</result_code>
4   <result_msg></result_msg>
5   <total_list_count>4</total_list_count>
6   <devices>
7     <device>
8       <device_Id>00000004dd2544ffff03777</device_Id>
9       <device_Name>로라_시험망</device_Name>
10      <device_type>070</device_type>
11      <model_Type>미지정</model_Type>
12      <model_Name>미지정</model_Name>
13      <manufacturer_Name>미지정</manufacturer_Name>
14      <category_Id>001</category_Id>
15      <category_Name>외식/유통</category_Name>
16      <location_Alt>37.573494400935964</location_Alt>
17      <location_Lon>126.98375701904297</location_Lon>
18      <location_Addr>대한민국 서울특별시 종로구 인사동9길 26</location_Addr>
19      <status>D02003</status>
20      <fault_Yn>N</fault_Yn>
21      <alive_Yn>Y</alive_Yn>
22      <public_Yn>Y</public_Yn>
23      <discovery_Yn>Y</discovery_Yn>
24      <owner_Id>daniel</owner_Id>
25      <category_Img_Link>/images/cate_img/cate_img01.png</category_Img_Link>
26      <regi_Date>2017-March-28th</regi_Date>
27    </device>
28  </devices>
29 </ThingPlug>
```

ThingPlug에 등록된
Device ID를 확인할 수 있다.

» 최신 데이터 확인

History

Collections

All Me Team

ThingPlug(00000004dd2544ffff0...
9 requests

01. Login

PUT Get Userkey

02. Device List

GET Get Device List

03. App -> ThingPlug

GET Get latest Data

PUT Device Control

GET Get Control Result

04. Data Subscription

POST sub create (Data)

GET sub retrieve (Data)

PUT sub update (Data)

DEL sub delete (Data)

▶ Get latest Data		
GET	https://onem2m.skot.com:9443/0000000000000004/v1_0/remoteCSE-00000004dd2544ffff03...	Params
Authorization Headers (4) Body Pre-request Script Tests		
	Key	Value
<input type="checkbox"/>	Accept	application/json
<input checked="" type="checkbox"/>	X-M2M-RI	00000004dd2544ffff03777_0002
<input checked="" type="checkbox"/>	X-M2M-Origin	00000004dd2544ffff03777
<input checked="" type="checkbox"/>	ukey	ZG9qcUNscTA3RlhLZ1IsbHB6ZUNPcWRUMml
	New key	value

- 디바이스의 가장 최신 데이터를 요청한다.

>> 최신 데이터 확인

<Req>

GET /<APP EUI>/v1_0/remoteCSE-<LTID>/container-LoRa/latest HTTP/1.1
Host: onem2m.sktiot.com:9443
X-M2M-RI: ri
X-M2M-Origin: origin
ukey: <user key>

<Resp>

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3   <ty>4</ty>
4   <ri>CI000000000000071336306</ri>
5   <rn>CI000000000000071336306</rn>
6   <pi>CT00000000000000004847</pi>
7   <ct>2017-04-06T20:08:04+09:00</ct>
8   <lt>2017-04-06T20:07:58+09:00</lt>
9   <et>2017-04-07T20:08:04+09:00</et>
10  <st>354</st>
11  <cr>RC0000000000000382125</cr>
12  <cnf>LoRa/Sensor</cnf>
13  <cs>4</cs>
14  <con>1234</con>
15 </m2m:cin>
```

디바이스에 보낸 데이터를
<con> 에서 확인할 수 있다.

>> 디바이스 제어하기

History

Collections

All Me Team

ThingPlug(00000004dd2544ffff0...
9 requests

01. Login
PUT Get Userkey

02. Device List
GET Get Device List

03. App -> ThingPlug
GET Get latest Data
PUT Device Control
GET Get Control Result

04. Data Subscription
POST sub create (Data)
GET sub retrieve (Data)
PUT sub update (Data)
DEL sub delete (Data)

Device Control

PUT https://onem2m.skot.com:9443/0000000000000004/v1_0/mgmtCmd-00000004dd2544ffff03... Params

Authorization Headers (4) Body Pre-request Script Tests

Key	Value
X-M2M-RI	00000004dd2544ffff03777_0012
X-M2M-Origin	00000004dd2544ffff03777
ukey	ZG9qcUNscTA3RIhLZ1IsbHB6ZUNPcWRUMml
Content-Type	application/xml

Authorization Headers (4) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary XML (application/xml)

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <m2m:m2m
3   xmlns:m2m="http://www.onem2m.org/xml/protocols"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xsi:type="ext"
6   <extra>0102</extra>
7 </m2m:m2m>
```

- 디바이스에 제어 명령을 보낼 수 있다.
- DevReset / extDevMgmt / RepPerchange / ReplImmediate

>> 디바이스 제어하기

<Req>

PUT /<APP EUI>/v1_0/mgmtCmd-<LTID>_extDevMgmt HTTP/1.1

Host: onem2m.sktiot.com:9443

X-M2M-RI: ri

X-M2M-Origin: origin

ukey: <user key>

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>

<m2m:mgc

xmlns:m2m="http://www.onem2m.org/xml/protocols"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<exe>true</exe>

<extra>0102</extra>

</m2m:mgc>

>> 디바이스 제어하기

<Resp>

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <m2m:mgc xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3   <ty>12</ty>
4   <ri>MC00000000000000007856</ri>
5   <rn>00000004dd2544fffff03777_extDevMgmt</rn>
6   <pi>CB00000000000000000014</pi>
7   <ct>2017-03-27T14:41:31+09:00</ct>
8   <lt>2017-03-27T14:41:31+09:00</lt>
9   <cmt>extDevMgmt</cmt>
10  <exe>>false</exe>
11  <ext>ND00000000000000003445</ext>
12  <m2m:exin>
13    <ty>8</ty>
14    <ri>EI000000000000000096615</ri>
15    <rn>EI000000000000000096615</rn>
16    <pi>MC00000000000000007856</pi>
17    <ct>2017-04-07T10:46:46+09:00</ct>
18    <lt>2017-04-07T10:46:46+09:00</lt>
19    <et>2017-04-08T10:46:46+09:00</et>
20    <exs>2</exs>
21    <ext>ND00000000000000003445</ext>
22    <extra>0102</extra>
23  </m2m:exin>
24 </m2m:mgc>
```

제어 명령 Instance ID,
제어 결과 조회할 때 사용한다.

"extDevMgmt" 에는 사용자 프로토콜을
사용할 수 있다.
Data type: hexa string

» 디바이스 제어 결과 확인하기

History

Collections

All Me Team

ThingPlug(00000004dd2544ffff0...
9 requests

01. Login

PUT Get Userkey

02. Device List

GET Get Device List

03. App -> ThingPlug

GET Get latest Data

PUT Device Control

GET Get Control Result

04. Data Subscription

POST sub create (Data)

GET sub retrieve (Data)

PUT sub update (Data)

DEL sub delete (Data)

Get Control Result	
GET	https://onem2m.sktiot.com:9443/0000000000000004/v1_0/mgmtCmd-00000004dd2544ffff03777_extDevMgmt/executeInstance- E1000000000000000096606
Authorization	Params
Headers (5)	Body
Pre-request Script	Tests
Key	Value
<input type="checkbox"/> Accept	application/xml
<input checked="" type="checkbox"/> X-M2M-RI	00000004dd2544ffff03777_0012
<input checked="" type="checkbox"/> X-M2M-Origin	00000004dd2544ffff03777
<input checked="" type="checkbox"/> ukey	ZG9qcUNscTA3RlhLZ1lsbHB6ZUNPcWRUMmly
<input checked="" type="checkbox"/> Content-Type	application/json;ty=12

- 제어 명령의 Instance ID로 결과를 확인할 수 있다.

>> 디바이스 제어 결과 확인하기

<Req>

GET /<APP EUI>/v1_0/mgmtCmd-<LTID>_extDevMgmt/execInstance-<EXEC INS> HTTP/1.1
Host: onem2m.sktiot.com:9443
X-M2M-RI: ri
X-M2M-Origin: origin
ukey: <user key>

<Resp>

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <m2m:exin xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3   <ty>8</ty>
4   <ri>EI000000000000000096615</ri>
5   <rn>EI000000000000000096615</rn>
6   <pi>MC00000000000000007856</pi>
7   <ct>2017-04-07T10:46:46+09:00</ct>
8   <lt>2017-04-07T10:46:49+09:00</lt>
9   <et>2017-04-08T10:46:46+09:00</et>
10  <exs>3</exs>
11  <exr>0</exr>
12  <ext>ND000000000000000003445</ext>
13  <extra>0102</extra>
14 </m2m:exin>
```

<exs>3</exs> : Finished
<exr>0</exr> : Success

>> 제어 결과 응답 코드

<exr>

exr	Execute Result	장치 제어 결과	
		Value	Interpretation
		0	STATUS_REQUEST_SUCCEEDED
		1	STATUS_REQUEST_UNSUPPORTED
		2	STATUS_REQUEST_DENIED
		3	STATUS_CANCELLATION_DENIED
		4	STATUS_INTERNAL_ERROR
		5	STATUS_INVALID_ARGUMENTS
		6	STATUS_RESOURCES_EXCEEDED
		7	STATUS_FILE_TRANSFER_FAILED
		8	STATUS_FILE_TRANSFER_SERVER_AUTHENTICATION_FAILURE
		9	STATUS_UNSUPPORTED_PROTOCOL
		10	STATUS_UPLOAD_FAILED
		11	STATUS_FILE_TRANSFER_FAILED_MULTICAST_GROUP_UNABLE_JOIN
		12	STATUS_FILE_TRANSFER_FAILED_SERVER_CONTACT_FAILED
		13	STATUS_FILE_TRANSFER_FAILED_FILE_ACCESS_FAILED
		14	STATUS_FILE_TRANSFER_FAILED_DOWNLOAD_INCOMPLETE
		15	STATUS_FILE_TRANSFER_FAILED_FILE_CORRUPTED
		16	STATUS_FILE_TRANSFER_FILE_AUTHENTICATION_FAILURE
		17	STATUS_FILE_TRANSFER_FAILED
		18	STATUS_FILE_TRANSFER_SERVER_AUTHENTICATION_FAILURE
		19	STATUS_FILE_TRANSFER_WINDOW_EXCEEDED
		20	STATUS_INVALID_UUID_FORMAT
		21	STATUS_UNKNOWN_EXECUTION_ENVIRONMENT
		22	STATUS_DISABLED_EXECUTION_ENVIRONMENT
		23	STATUS_EXECUTION_ENVIRONMENT_MISMATCH
		24	STATUS_DUPLICATE_DEPLOYMENT_UNIT
		25	STATUS_SYSTEM_RESOURCES_EXCEEDED
		26	STATUS_UNKNOWN_DEPLOYMENT_UNIT
		27	STATUS_INVALID_DEPLOYMENT_UNIT_STATE
		28	STATUS_INVALID_DEPLOYMENT_UNIT_UPDATE_DOWNGRADE_DISALLOWED
		29	STATUS_INVALID_DEPLOYMENT_UNIT_UPDATE_UPGRADE_DISALLOWED
		30	STATUS_INVALID_DEPLOYMENT_UNIT_UPDATE_VERSION_EXISTS
		31	STATUS_DEVICE_MANAGEMENT_TIME_OUT
		32	STATUS_NW_SVR_TRANSFER_FAILED

<exs>

exs	Execute Status	장치 제어 상태 정보	
		Value	Interpretation
		1	INITIATED
		2	PENDING
		3	FINISHED
		4	CANCELLING
		5	CANCELLED
		6	STATUS_NON_CANCELLABLE

≫ 데이터 Subscription

History

Collections

All Me Team

ThingPlug(00000004dd2544ffff0...
9 requests

01. Login

PUT Get Userkey

02. Device List

GET Get Device List

03. App -> ThingPlug

GET Get latest Data

PUT Device Control

GET Get Control Result

04. Data Subscription

POST sub create (Data)

GET sub retrieve (Data)

PUT sub update (Data)

DEL sub delete (Data)

sub create (Data)

POST https://onem2m.skiot.com:9443/0000000000000004/v1_0/remoteCSE-00000004dd2544ffff03... Params

Authorization Headers (6) Body Pre-request Script Tests

Key	Value
Accept	application/xml
X-M2M-RI	00000004dd2544ffff03777_00012
X-M2M-Origin	00000004dd2544ffff03777
uKey	ZG9qcUNscTA3RIhLZ1IsbHB6ZUNPcWRUMmly
X-M2M-NM	test_sub
Content-Type	application/vnd.onem2m-res+xml;ty=23

Authorization Headers (6) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary Text

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <m2m:sub
3   xmlns:m2m="http://www.onem2m.org/xml/protocols"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   <enc>
6     <rss>1</rss>
7   </enc>
8   <nu>HTTP|http://[redacted]:5000</nu>
9   <nct>2</nct>
10 </m2m:sub>
```

- 데이터가 생성되었을 때 지정된 서버로 Push 설정
- Trigger를 활용해 조건을 설정할 수 있다.

>> 데이터 Subscription

<Req>

POST /<APP EUI>/v1_0/remoteCSE- <LTID>/container-LoRa HTTP/1.1
Host: onem2m.sktiot.com:9443
X-M2M-RI: ri
X-M2M-Origin: origin
ukey: <user key>
X-M2M-NM: test_sub
Content-Type: application/vnd.onem2m-res+xml;ty=23

```
<?xml version="1.0" encoding="UTF-8"?>
<m2m:sub
  xmlns:m2m="http://www.onem2m.org/xml/protocols"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <enc>
    <rss>1</rss>
  </enc>
  <nu>HTTP|http://xxx.xxx.xxx.xxx:5000</nu>
  <nct>2</nct>
</m2m:sub>
```



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
  <m2m:sub xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
    <t>23</t>  
    <r>SS00000000000000000003876</r>  
    <rn>test_sub</rn>  
    <p>CT00000000000000000004847</pi>  
    <ct>2017-04-07T16:46:40+09:00</ct>  
    <lt>2017-04-07T16:46:40+09:00</lt>  
    <enc>  
      ...  
      <rss>1</rss>  
    </enc>  
    <nu>HTTP|http://[redacted]:5000</nu>  
    <nct>2</nct>  
  </m2m:sub>
```

App Server의 IP:Port 가 등록되었으면 성공

≫ 데이터 Subscription 동작 확인

```
*COM10@38400,8,n,1 - Token2Shell
Session View Tools Help Documents
10,38400 Serial(COM) Port
COM10@38400,8,n,1 x
> COM Port : COM10
> Baud Rate : 38400
> Data Bits : 8
> Parity : None
> Stop Bits : 1
> Flow Control : None
Connected to COM10.
Send Data
Send Data Port:1, Byte :5
Event mode
txDone
OnRadioTxDone
RSSI -62, SNR 39
RX DONE
Line Sender
AT+SEND 011234567890
```



UDP Setup | Serial | TCP Client | TCP Server | UDP | Test Mode | About

Received data

POST / HTTP/1.1
X-M2M-Origin: 00000004dd2544ffff03777
X-M2M-RI: ad412320-a155-4b3a-bddd-9d3712ea9b9e
Accept: application/xml
Content-Type: application/vnd.onem2m-ntfy+xml; charset=UTF-8
User-Agent: Jakarta Commons-HttpClient/3.0.1
Host: 2[REDACTED]4:5000
Content-Length: 581

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><ty>4
</ty><ri>C10000000000071388485</ri><rn>
C10000000000071388485</rn><pi>CT00000000000000004847</pi>
<ct>2017-04-07T17:07:10+09:00</ct><lt>2017-04-07T17:07:10+
09:00</lt><sr>/0000000000000004/v1_Q/remoteCSE-
00000004dd2544ffff03777/container-LoRa/subscription-test_sub
</sr><et>2017-04-08T17:07:10+09:00</et><st>364</st><cr>
RC0000000000000382125</cr><cnf>LoRa/Sensor</cnf><cs>10
</cs><con>1234567890</con></m2m:cin>

Server status
Port
5000 [Close]

TEA authorization
TEA key
1: 01020304 3: 090A080C
2: 05060708 4: 0D0E0F10
☐ Client authorization

Client connection status
오후 5:07:21: 61,250,21,79 Client
Clients count: 0

>> 데이터 포맷

<Data>

POST / HTTP/1.1

X-M2M-Origin: 00000004dd2544ffff03777

X-M2M-RI: ad412320-a155-4b3a-bddd-9d37f2ea9b9e

Accept: application/xml

Content-Type: application/vnd.onem2m-ntfy+xml;charset=UTF-8

User-Agent: Jakarta Commons-HttpClient/3.0.1

Host: xxx.xxx.xxx.xxx:5000

Content-Length: 581

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<m2m:cin

xmlns:m2m="http://www.onem2m.org/xml/protocols"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<ty>4</ty>

<ri>CI000000000000071388485</ri>

<rn>CI000000000000071388485</rn>

<pi>CT00000000000000004847</pi>

<ct>2017-04-07T17:07:10+09:00</ct>

<lt>2017-04-07T17:07:10+09:00</lt>

<sr>/0000000000000004/v1_0/remoteCSE-00000004dd2544ffff03777/container-LoRa/subscription-test_sub</sr>

<et>2017-04-08T17:07:10+09:00</et>

<st>364</st>

<cr>RC000000000000000382125</cr>

<cnf>LoRa/Sensor</cnf>

<cs>10</cs>

<con>1234567890</con>

</m2m:cin>



Python Client

>> Python MQTT Client 사용방법 익히기



<https://github.com/kaizen8501/thingplugAPI>

MQTT Client

MQTT 프로토콜을 사용해서
ThingPlug의 Subscription을 사용하는 예제

> 실습의 편의를 위해 exe로 변환해서 테스트

>> Python MQTT Client 사용방법 익히기

강의자료\Utility\03_cli_command

이름	수정된 날짜	유형	크기
01_login.exe	2017-04-13 오후...	응용 프로그램	6,459KB
02_get_device_list.exe	2017-04-13 오후...	응용 프로그램	6,459KB
03_mgmtCmd.exe	2017-04-14 오후...	응용 프로그램	6,460KB
04_get_latest_data.exe	2017-04-13 오후...	응용 프로그램	6,462KB
05_subscription_mqtt.exe	2017-04-13 오후...	응용 프로그램	6,460KB

- 01_login.exe** : ThingPlug 포털의 ID와 PWD로 사용자 인증키를 요청한다.
- 02_get_device_list.exe** : ThingPlug 계정에 등록된 디바이스 목록을 요청한다.
- 03_mgmtCmd.exe** : 디바이스에 디바이스 리셋 등의 제어 명령을 보낸다.
- 04_get_latest_data.exe** : 디바이스의 가장 최신 데이터를 요청한다.
- 05_subscription_mqtt.exe** : 디바이스의 데이터를 MQTT로 PUSH 받는다.

>> User Key를 확인 한다.

사용 방법

usage: 01_login.exe [-h] -u USER_ID -p USER_PW [-th THINGPLUG_HOST] [-tp THINGPLUG_PORT]

-u USER_ID	ThingPlug User ID
-p USER_PW	ThingPlug User Password
-th THINGPLUG_HOST	ThingPlug Host IP(Default:onem2m.sktiot.com)
-tp THINGPLUG_PORT	ThingPlug Port(Default:9443)

사용 예제

```
$ 01_login.exe -u USER_ID -p USER_PASSWORD
```

≫≫ 디바이스 리스트를 확인 한다.

사용 방법

usage: 02_get_device_list.exe [-h] -u USER_ID -p USER_PW [-th THINGPLUG_HOST] [-tp THINGPLUG_PORT] [-ae APP_EUI]

-u USER_ID	ThingPlug User ID
-p USER_PW	ThingPlug User Password
-th THINGPLUG_HOST	ThingPlug Host IP(Default:onem2m.sktiot.com)
-tp THINGPLUG_PORT	ThingPlug Port(Default:9443)
-ae APP_EUI	ThingPlug APP EUI(Default:ThingPlug)

사용 예제

```
$ 02_get_device_list.exe -u USER_ID -p USER_PASSWORD
```


≫ 디바이스에 제어 명령을 보낸 후 결과를 확인한다.

사용 방법

```
usage: 03_mgmtCmd.exe [-h] -u USER_ID -p USER_PW [-th THINGPLUG_HOST] [-tp THINGPLUG_PORT]
                        [-ae APP_EUI] -ni NODE_ID -cmt COMMAND_TYPE [-extra REQUEST_ARGUMENT]
```

-u USER_ID	ThingPlug User ID
-p USER_PW	ThingPlug User Password
-ni NODE_ID	ThingPlug Node ID
-th THINGPLUG_HOST	ThingPlug Host IP(Default:onem2m.sktiot.com)
-tp THINGPLUG_PORT	ThingPlug Port(Default:9443)
-ae APP_EUI	ThingPlug APP EUI(Default:ThingPlug)
-cmt COMMAND_TYPE	ThingPlug Command Type. (DevReset, extDevMgmt, ReplImmediate, RepPerChange)
-extra REQUEST_ARGUMENT	ThingPlug Execute Request Argument

사용 예제

```
$ 03_mgmtCmd.exe -u USER_ID -p USER_PASSWORD -ae APP_EUI -ni xxxxxxx -cmt DevReset
```

➤➤ Polling 방식으로 최신 데이터를 조회한다.

사용 방법

usage: 04_get_latest_data.exe [-h] -u USER_ID -p USER_PW -ni NODE_ID -ct CONTAINER [-th THINGPLUG_HOST]
[-tp THINGPLUG_PORT] [-ae APP_EUI] [-pt PERIOD_TIME] [-el ENABLE_LOG]

-u USER_ID	ThingPlug User ID
-p USER_PW	ThingPlug User Password
-ni NODE_ID	ThingPlug Node ID
-ct CONTAINER	ThingPlug Container Name
-th THINGPLUG_HOST	ThingPlug Host IP(Default:onem2m.sktiot.com)
-tp THINGPLUG_PORT	ThingPlug Port(Default:9443)
-ae APP_EUI	ThingPlug APP EUI(Default:ThingPlug)
-pt PERIOD_TIME	Get Latest Data Period Time(Default:0,One Time)
-el ENABLE_LOG	ENABLE_LOG

사용 예제

```
$ 04_get_latest_data.exe -u USER_ID -p USER_PASSWORD -ae APP_EUI -ni xxxxxxx -ct LoRa
```

≫ Subscription을 수행하는 MQTT Client

사용 방법

usage: 05_subscription_mqtt.exe [-h] -u USER_ID -p USER_PW [-ni NODE_ID] [-ct CONTAINER] [-th THINGPLUG_HOST] [-tp THINGPLUG_PORT] [-ae APP_EUI] [-ci MQTT_CLIENT_ID] [-el ENABLE_LOG]

-u USER_ID	ThingPlug User ID
-p USER_PW	ThingPlug User Password
-ni NODE_ID	ThingPlug Node ID
-ct CONTAINER	ThingPlug Container Name
-th THINGPLUG_HOST	ThingPlug Host IP(Default:onem2m.sktiot.com)
-tp THINGPLUG_PORT	ThingPlug Port(Default:9443)
-ae APP_EUI	ThingPlug APP EUI(Default:ThingPlug)
-ci MQTT_CLIENT_ID	ThingPlug MQTT Client ID(Default:bridge)
-el ENABLE_LOG	ENABLE_LOG

사용 예제

\$ 05_subscription_mqtt.exe -u *USER_ID* -p *USER_PASSWORD* -ae *APP_EUI* -ni *xxxxxxxx* -ct *LoRa*

Q&A 감사합니다
