ThingPlug 중심의 IoT 서비스 개발

>> loT 오픈하우스 소개

역 할

- SKT New ICT 활성화를 위하여 개발자 및 스타트업의 상품 개발부터 서비스의 상용화까지 종합적인 지원 제공
- LTE Cat-M1, LoRa 개발 테스트 환경 지원 및 정기 강의 제공

지원대상 및 신청방법

- 지원대상 : LTE Cat-M1, LoRa 서비스를 기획 및 개발중인 다양한 기업
- 신청방법 : IoT Portal 내 IoT OpenHouse (https://www.sktiot.com)

지원 업무

- 서비스의 기획부터 상용화까지 컨설팅 지원
- 서비스 기술 지원 및 테스트 환경 제공
- Device 및 플랫폼 연동 기술 상담
- 정규 강의 제공



>> loT 오픈하우스 지원 내용

컨설팅



- Cat-M1 서비스 아이디어
- Cat-M1 서비스 상용화 프로세스 지원

개발지원



- Cat-M1 서비스 개발 지원
- 테스트 공간 및 장비 지원
- IoT 플랫폼 연동 지원
- 온라인 서비스를 통한 지원

교육

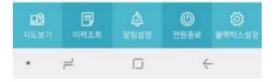


- 분기별 정규 강의 진행
- 필요 시 추가 강의를 편성하여 개별 강의 진행

IOT 프로젝트

loT 블랙박스 블랙박스에 Cat.M1 블랙박스용 전용 모뎀을 연동하여, 고객이 휴대폰으로 주차 차량의 이벤트 등을 확인할 수 있는 서비스 [SKT Role] Cat.M1 ((tp)) LTE 구성도 Network 블랙박스 Cat.M1 블랙박스 제조사 B2C유저 Smart 스마트폰 차량용통신모뎀 Fleet P/F 자체 App. 서버 · 주차 위치파악/충격 감지 시 이벤트 Noti. 및 사진/GPS 전송 특징 ·사고 통보 및 짧은 영상 전송, 차량 도난감지 및 견인 알림, 위급상황 알림, 배터리 방전 보호 알림 기능 •직접 블랙박스를 확인해야하는 고객의 불편함을 저렴한 기대효과 요금으로 해소 · App.을 통한 주차위치확인, 차량관리 및 차량 Safety 강화

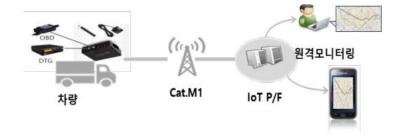




차량 관제(Car Tracking)

화물차/렌터카 등의 차량에 Cat.M1 통신모뎀 (+OBD)을 장착하고 차량의 위치/운행 정보를 수집 분석하여 제공

구성도



특징

- •운행 기록 실시간 전송, 실시간 차량관제(운전습관 등)
- ·차량 Lifecycle의 효과적/ 효율적 관리

기대효과

- ·차량플랫폼을 활용한 운행기록 분석을 통해 유류비, 보험료 등의 비용 절감
- •화물배송 및 배차관리 효율성 극대화



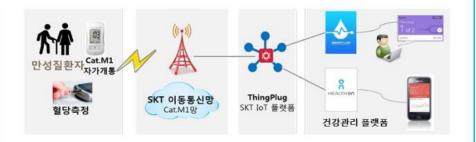




혈당 측정 관리

혈당측정기에 Cat.M1 통신과 헬스케어 플랫폼을 연동한 실시간 진단/관리 등 통합형 헬스케어 서비스

구성도



특징

- ·기존 Bluetooth/NFC 모델 대비 간편한 Data 수집
- ·Data 축적&Tracking 기능 제공으로 효율적인 건강관리

기대효과

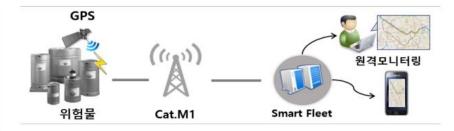
· 병원/보험사 연계를 통해 개인별 건강관리를 강화하고, 중증 질환을 사전적으로 예방하여 개인적/국가적 의료비용 절감



Asset Tracking

화학약품, 위험물 등에 GPS 추적이 가능한 Cat.M1 단말을 부착하여 주요자산 위치 및 사용현황을 관제하는 서비스

구성도



특징

- · 주요 동산자산(중요자산 or Movable Asset)의 활용도 및 실시간 위치 안내 서비스 제공
- ·배터리 용량에 따라 최소 3~5개월 이상 Tracking 가능

기대효과

· Movable Asset에 대한 효율적인 관리 감독 가능





자전거 잠금장치(Cat.M1)

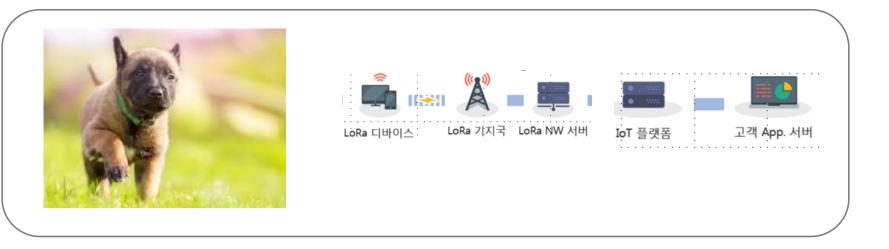
자전거 위치 추적과 충격 감지시 사용자에게 알림



데이터: 자전거의 위치와 충격 감지

반려동물 트래커(LoRa)

반려동물의 위치정보를 사용자의 스마트폰 앱으로 전송한다.



데이터: 반려동물의 위치정보

IoT 구명조끼(LoRa)

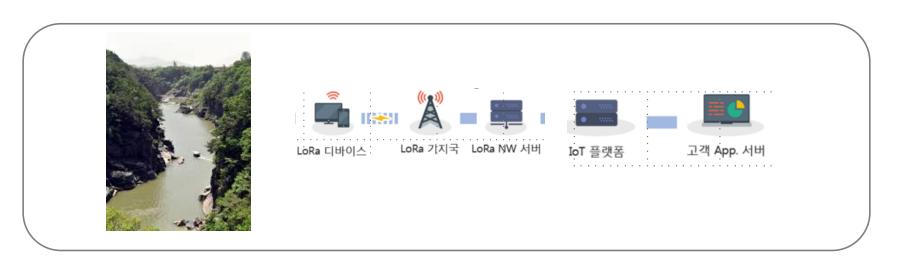
수상활동 중 발생할 수 있는 긴급 상황에 대비하여 사용자 상태 모니터링



데이터: 심박, 위치, 온도

하천 모니터링 시스템 (LoRa)

하천의 유속과 오염 정도를 수집하여 하천의 생태계 모니터링



데이터: 유속, 오염 정도

1. 누구나 한번 쯤 생각해 봤을만한 아이디어.

2. 모니터링/트래킹 서비스가 많다.

3. 전국 어디에서나 사용 가능하다.

4. 통신 빈도가 적다.

1. 누구나 한번 쯤 생각해 봤을만한 아이디어.

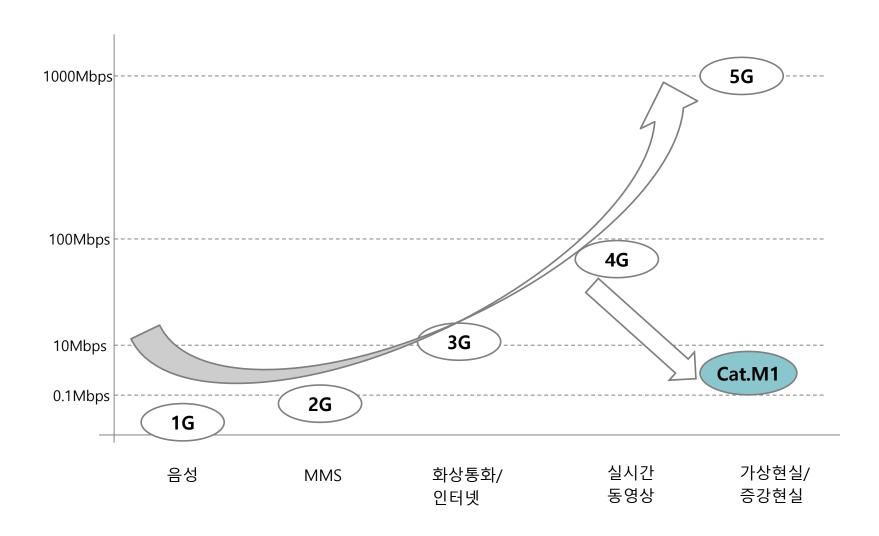
2. 모니터링/트래킹 서비스가 많다.

3. 전국 어디에서나 사용 가능하다.

4. 통신 빈도가 적다.

통신 단말과 통신 비용이 저렴해졌다.

>> Wireless Network



>> loT 플랫폼



INDEX

- **01** loT 플랫폼
- **02** ThingPlug
- **03** 단말 개발하기
- 04 서비스 개발하기
- 05 ThingPlug API 활용

IOT 플랫폼







대화할 수 없는 상황이면?



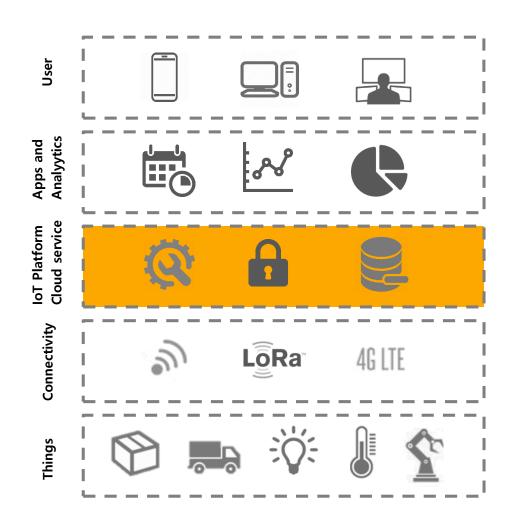
대화할 수 없는 상황이면?



디바이스와 사용자 사이에 항상 메시지를 전달할 수 있는 칠판을 둔다.

= IoT 플랫폼

>> loT 플랫폼



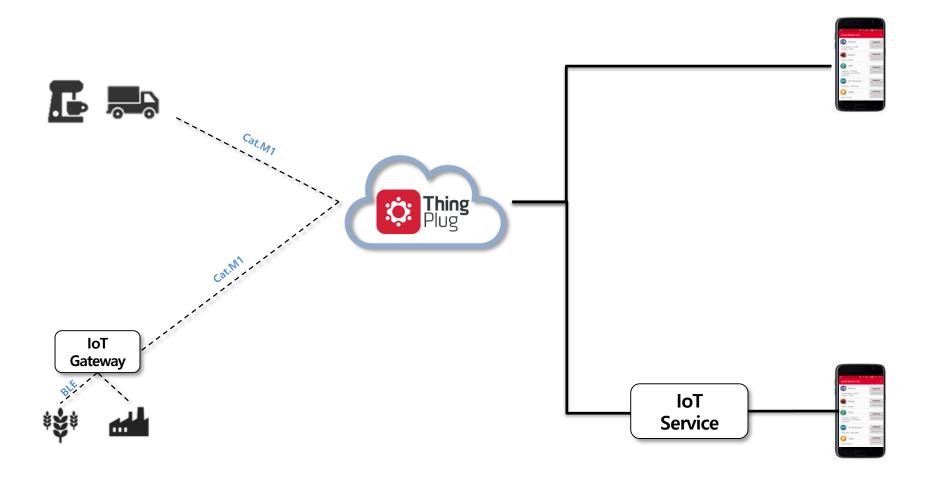
- 1. 다양한 connectivity 지원
- 2. 디바이스 관리
- 3. 데이터베이스
- 4. 데이터 처리 관리
- 5. 대시보드
- 6. 3rd-party 인터페이스 제공

>> loT 플랫폼



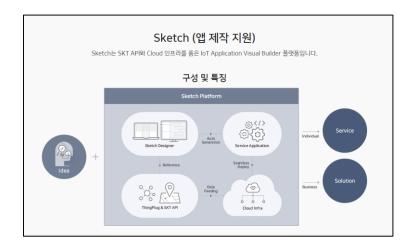
- 1. 다양한 connectivity 지원
- 2. 디바이스 관리
- 3. 데이터베이스
- 4. 데이터 처리 관리
- 5. 대시보드
- 6. 3rd-party 인터페이스 제공

>> IoT 디바이스부터 서비스까지



>> ThingPlug의 기능





통합 개발환경

앱 제작 지원

시각화 및 분석



API 및 개발 툴 지원

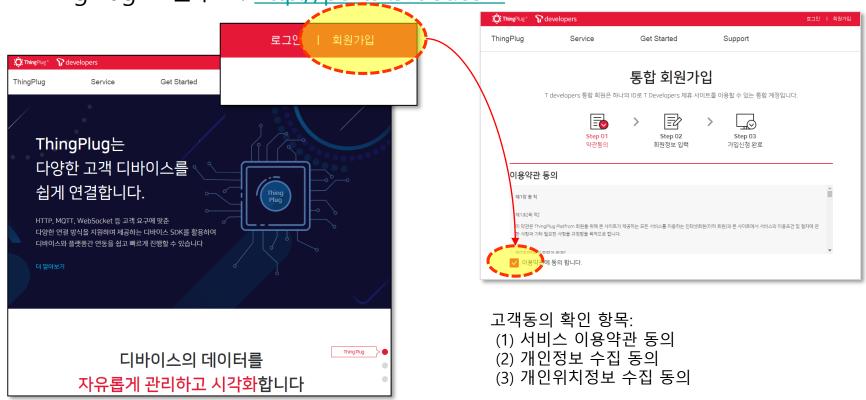






>> 회원 가입

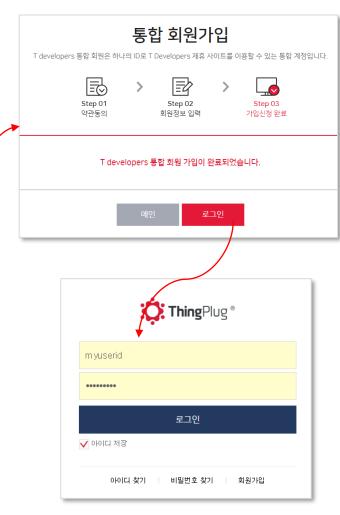
ThingPlug 포털주소: http://portal.sktiot.com



>> 회원 가입

• 사용자 정보 입력





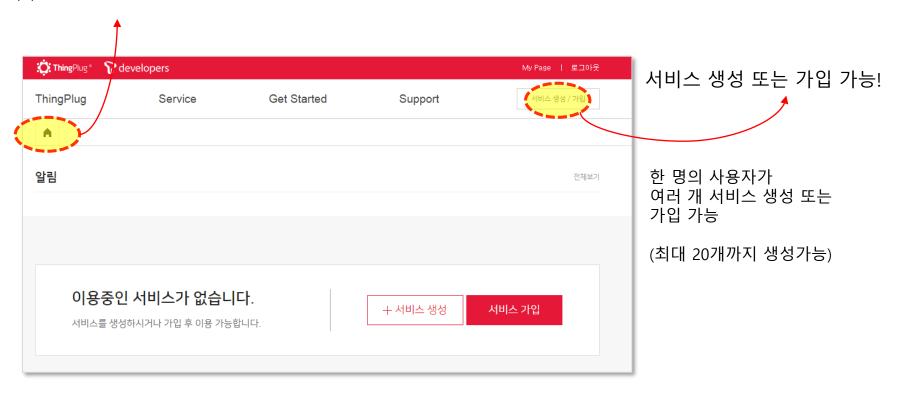
사용자 ID: myuserid



IoT 오픈 하우스

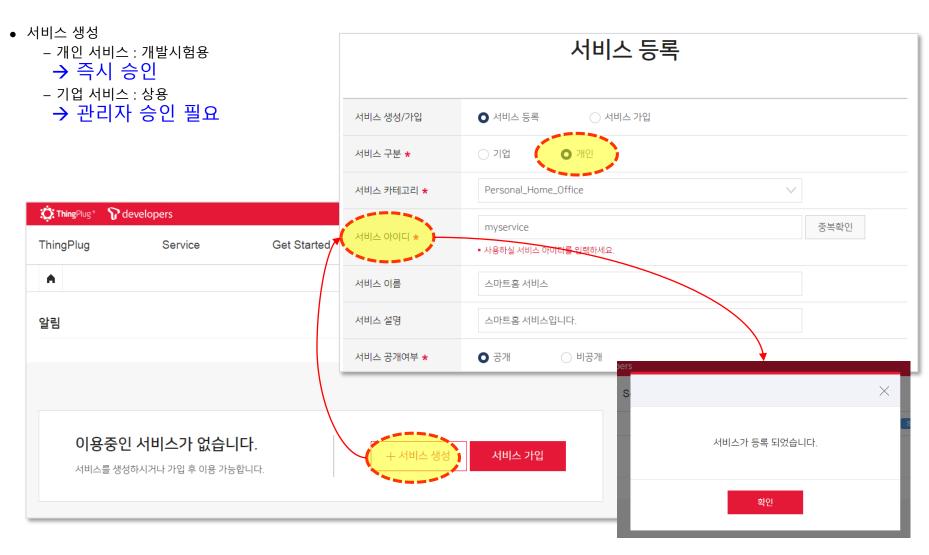
>> 서비스 생성

서비스 Home



IoT 오픈 하우스

>> 서비스 생성



>> 서비스 관리

서비스 상태	활성
과금 여부	비과금
최대 연결 디바이스 \star	30 개 ■ 최대 제공 연결 디바이스 수는 30개이며 변경 시 관리자 승인 후 변경됩니다.
최대 룰엔진 \star	10 개 ■ 최대 제공 <mark>물 엔진 수는 25개</mark> 이며 변경 시 관리자 승인 후 변경됩니다.
최대 회원가입 \star	30 명 ■ 최대 제공 가입 수는 100명이며 변경 시 관리자 승인 후 변경됩니다.
데이터 보관 기간 *	● 일 월 15 일 ■ 기본 제공 데이터 보관기간은 15일이며 변경 시 관리자 승인 후 변경됩니다. ■ 최대 데이터 보관 기간은 XXXXXX입니다.

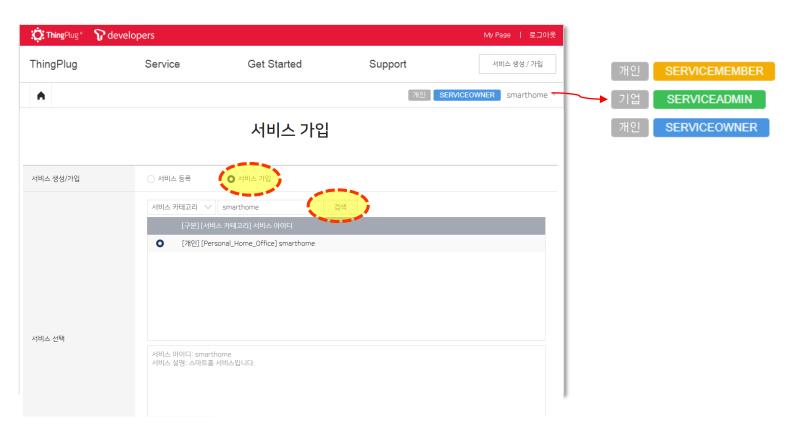
기업 서비스나 관리자 승인으로 리소스 제한을 변경할 수 있다.

>> 서비스 관리

타 서비스 가입 (공개 서비스에만 가입신청 가능)

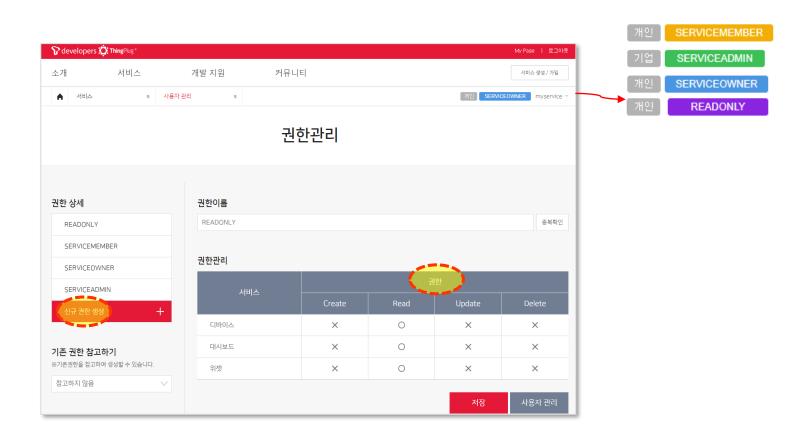
SERVICEMEMBER 권한을 갖게 됨

SERVICEOWNER에 의해SERVICEADMIN 권한으로 승격 가능



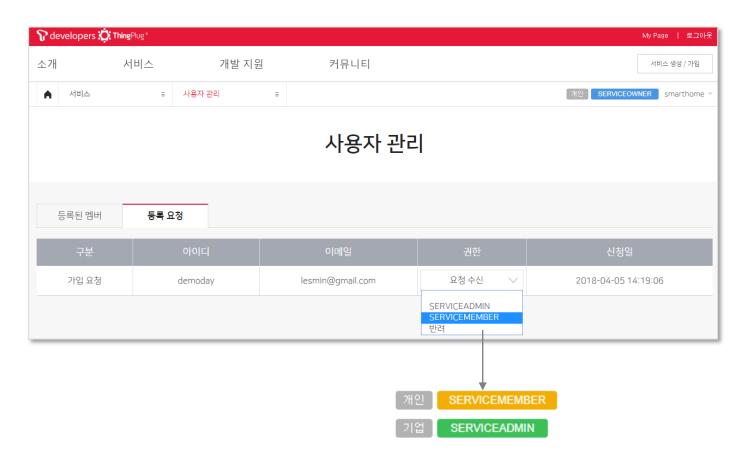
>> 서비스 관리

SERVICEOWNER 또는 SERVICEADMIN은 신규 권한도 생성 가능

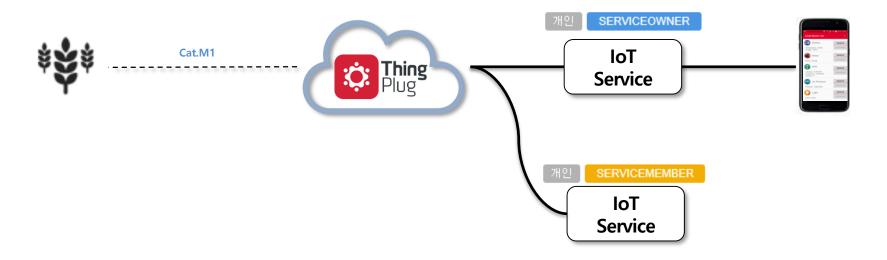


>> 서비스 관리

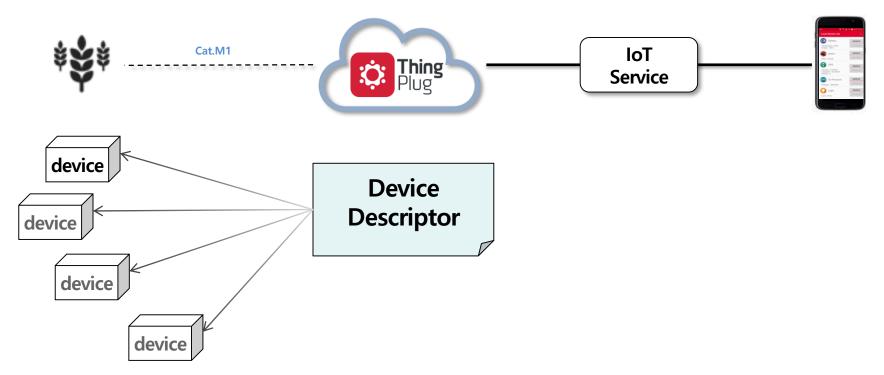
서비스 사용자 관리 (SERVICEOWNER 또는 SERVICEADMIN 만 가입신청 승인 가능) SERVICEMEMBER 또는 SERVICEADMIN 권한 부여하여 서비스 가입신청 승인



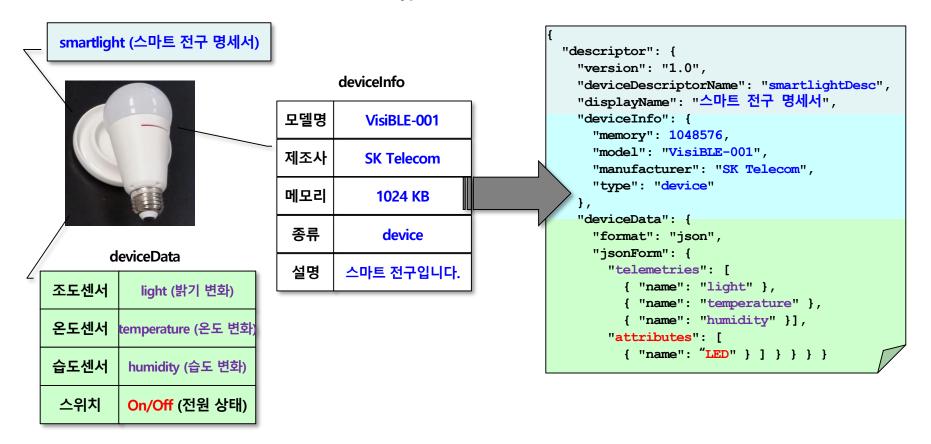
>> 서비스마다 권한을 설정할 수 있다





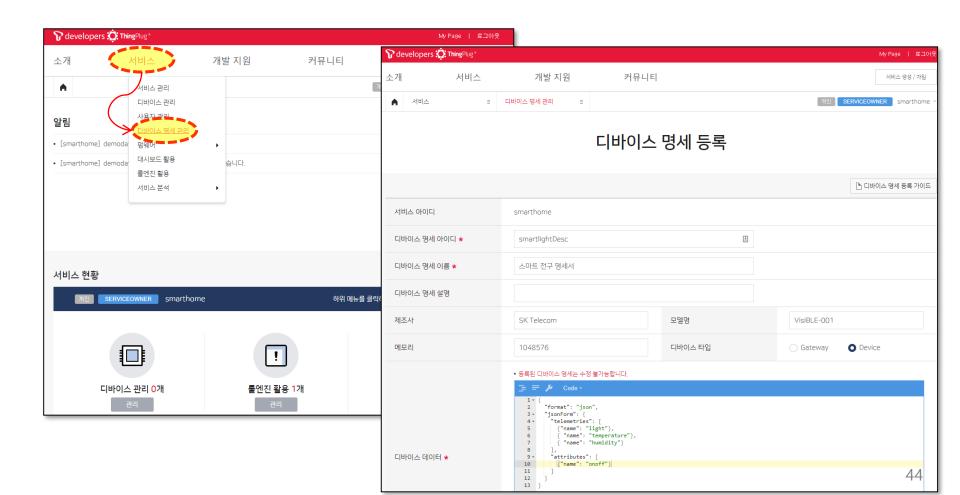


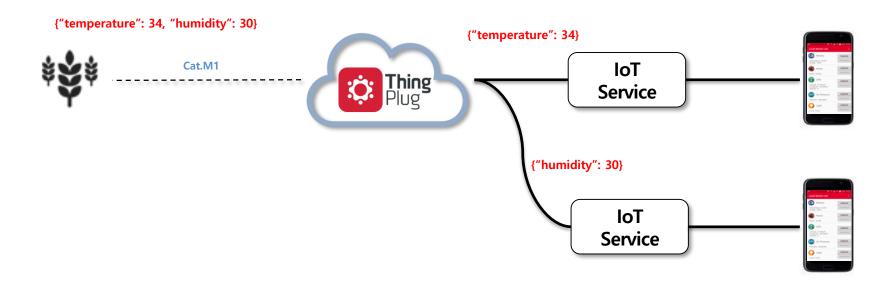
변화되는 값을 시간에 따라 관리하고 싶다면, 시계열 type인 'telemetry'로 정의! 현재의 최신 전원상태만 관리하고 싶다면, 속성 type인 'attribute'로 정의!



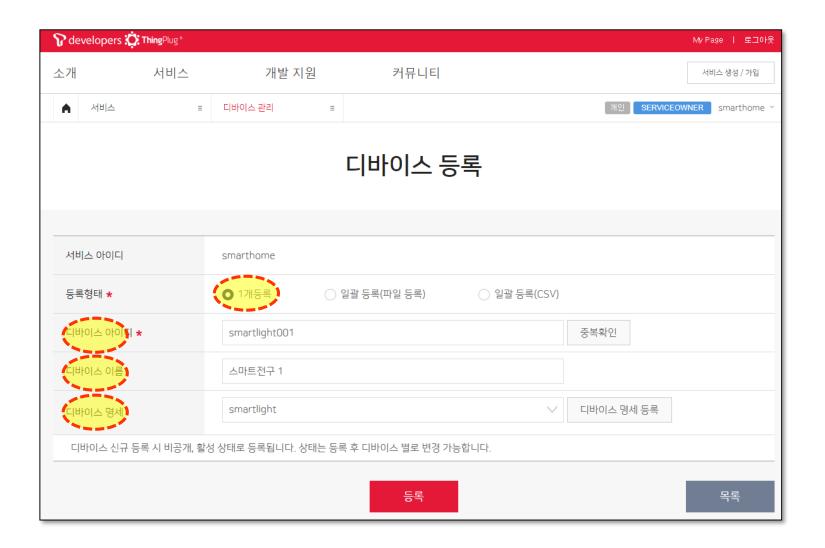
```
"format": "json",
"jsonForm": {
 "telemetries": [
    "name": "temperature",
    "tag": "temperature"
    "name": "humidity",
    "tag": "humidity"
 "attributes": [
    "name": "LED",
    "commandable": true
    "name": "lat",
    "tag": "latitude"
    "name": "long",
    "tag": "longitude"
```

디바이스(들)을 등록하기 위해서는, 디바이스 명세를 먼저 작성 해야함 디바이스 명세는, 디바이스의 기본적인 정보와 플랫폼에 전달할 Data 종류를 기술





>> 디바이스 관리



>> 디바이스 관리

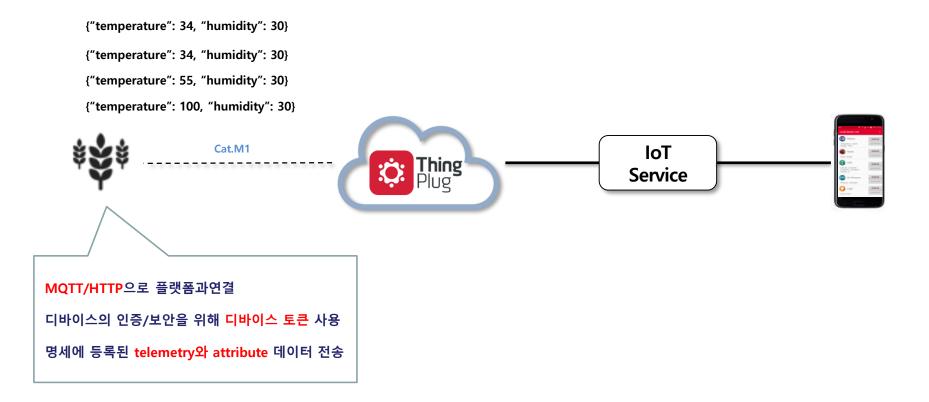
ThingPlug에서는 <mark>디바이스 아이디와 디바이스 토큰</mark>으로 단말을 관리한다. 단말이 ThingPlug에 접속하기 위해서는 <mark>디바이스 아이디와 토큰</mark>이 필요하다. 사용자가 단말의 데이터를 조회하기 위해서는 <mark>디바이스 아이디로</mark> 구분한다.



디바이스 명세 ??



>> ThingPlug와 연결되는 디바이스



>> ThingPlug에 데이터 보내기





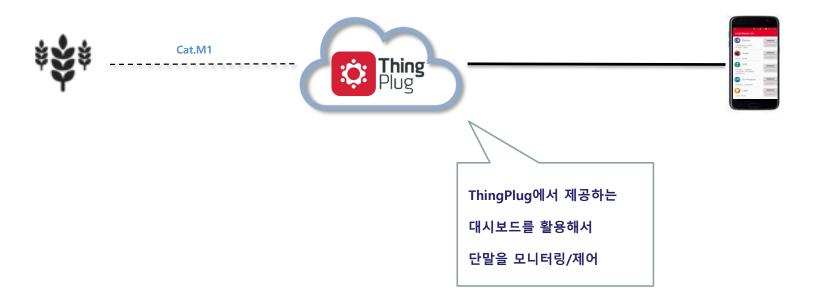
```
AT+CEREG? → 네트워크 상태 확인
+CEREG: 0,1
OK
AT*RNDISDATA=1
*RNDISDATA:1
OK
AT+SKTPCON=1,MQTT,test.sktiot.com,1883,120,1,simple_v1,<device_token>,<service_id>,<device_id>
OK
+SKTPCON=0
AT+SKTPDAT=1,telemetry,0,{"temperature":30, "humidity":42}
OK
```

서비스 개발하기

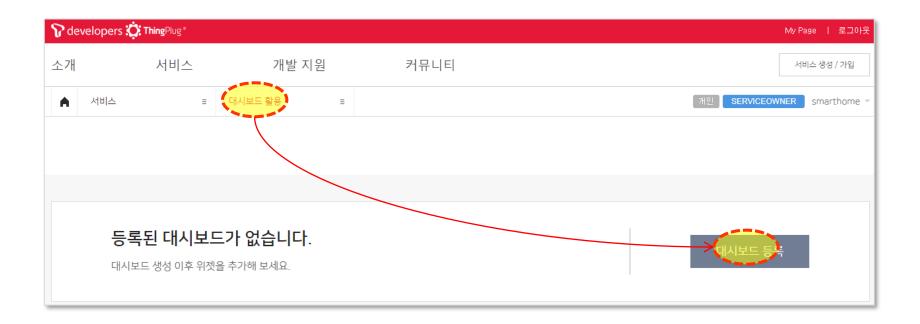
ThingPlug를 활용한 서비스 개발



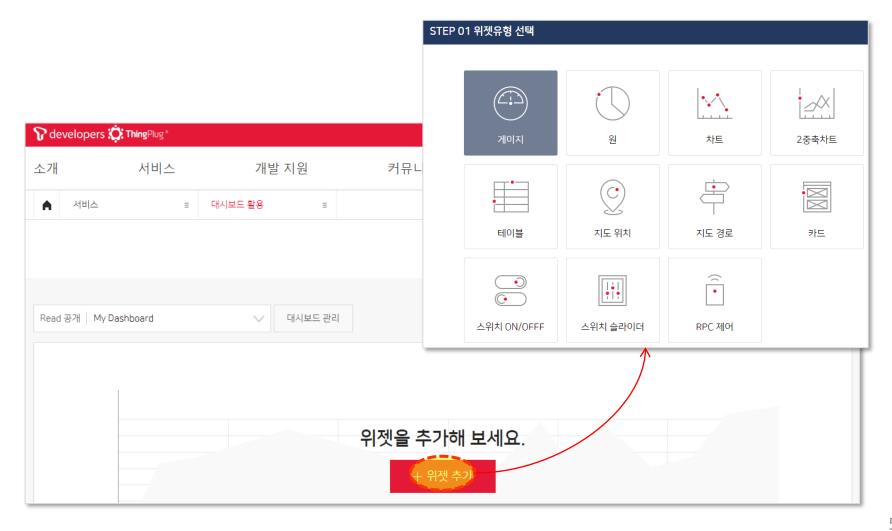
>> ThingPlug 대시보드 활용



>> ThingPlug 대시보드 활용



>> ThingPlug 대시보드 활용



>> 환경 모니터링

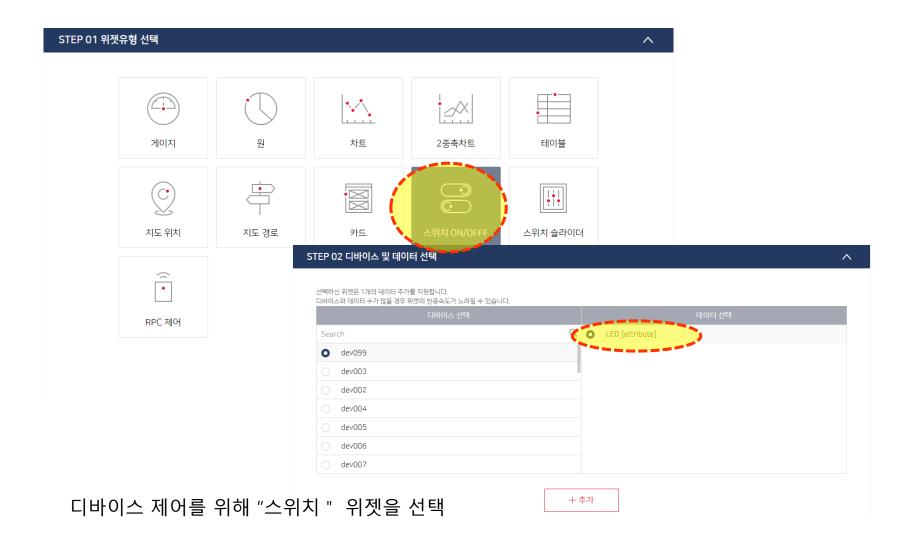


온도 변화를 차트로 표현하기 위해 "차트" 위젯에서 "temperature", "humidity"를 선택

>> 환경 모니터링

```
AT+SKTPDAT=1, telemetry, 0, {"temperature": 32, "humidity": 33}
                                                                  Cat.M1 디바이스의 데이터를 차트로 표현
AT+SKTPDAT=1, telemetry, 0, {"temperature": 32, "humidity": 33}
AT+SKTPDAT=1, telemetry, 0, {"temperature": 32, "humidity": 33}
AT+SKTPDAT=1, telemetry, 0, {"temperature": 32, "humidity": 33}
AT+SKTPDAT=1, telemetry, 0, {"temperature": 33, "humidity": 34}
AT+SKTPDAT=1, telemetry, 0, {"temperature": 33, "humidity": 34}
                                      온도변화
                                                                                                                       34
                                        30
                                                                                  15:20
                                                              15:15
                                                                                                       15:25
                                                          dev099.temperature  
dev099.humidity
```

>> 디바이스 제어



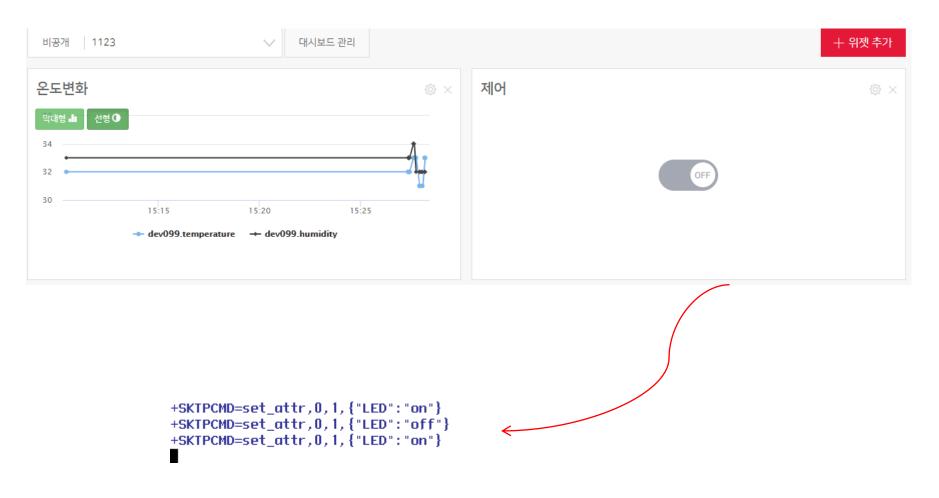
>> 디바이스 제어

STEP 03 옵션		^
위젯타이틀 *	제어	
제어요청값 *	○ 숫자	
 ① 디바이스와 데이터 수가 많을 경우 	정상적으로 위젯이 동작하지 않을 수 있습니다.	

스위치를 눌렀을 때 디바이스로 전달 할 메시지를 입력

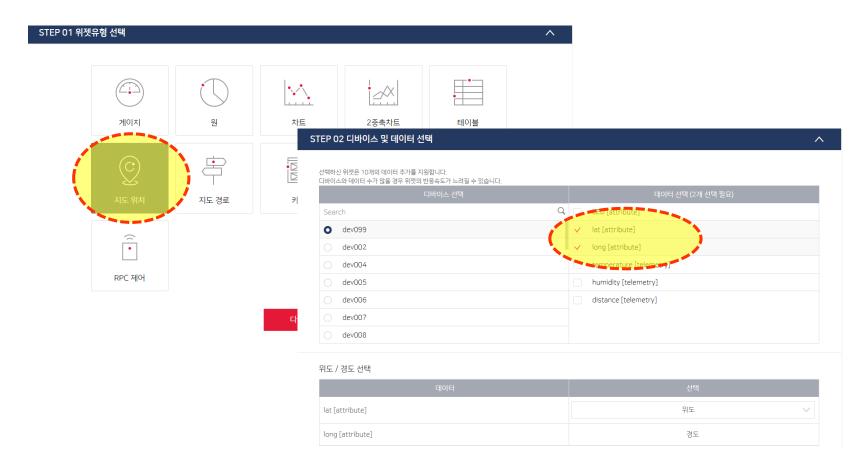
IoT 오픈 하우스

>> 디바이스 제어



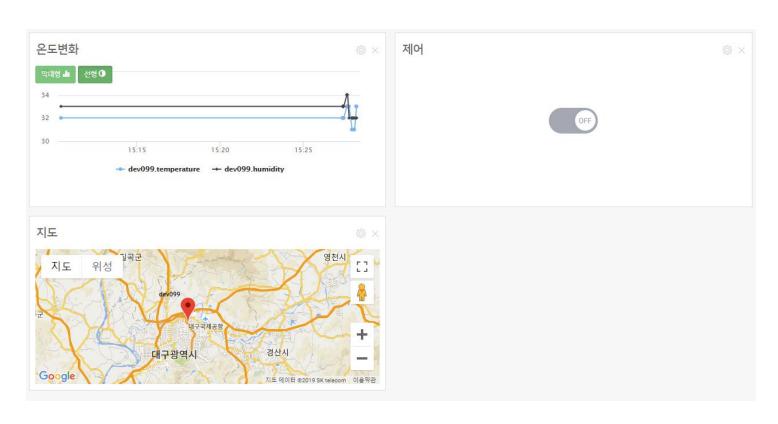
제어 메시지를 디바이스에서 확인

>> 위치 정보



위치정보를 지도에 표현하기 위해서 "지도" 위젯과 "lat", long" 데이터를 선택

>> 위치 정보

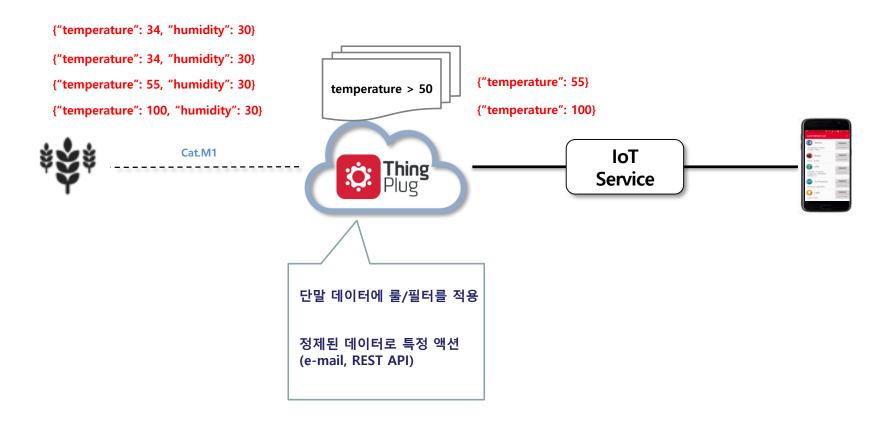


AT+SKTPDAT=1,attribute,0,{"lat":35.8928704, "long":128.5972525} OK

디바이스에서 위치 정보를 전송하면, 지도 위젯에 좌표가 찍힌다.



>> 룰 엔진



>> 룰 엔진

기본 필터

별다른 조건 없이 디바이스 데이터가 통과

디바이스 이름 필터

특정 디바이스를 선별하여 데이터(Telemetry, Attribute)를 정제하기 위해 사용

타입 필터

데이터의 조건(only Telemetry / only Attribute)으로 정제하기 위해 사용

숫자 필터

특정 Key값에 조건을 걸어 데이터를 정제하고자 할 때 사용

문자 필터

특정 Key값에 조건을 걸어 데이터를 정제하고자 할 때 사용

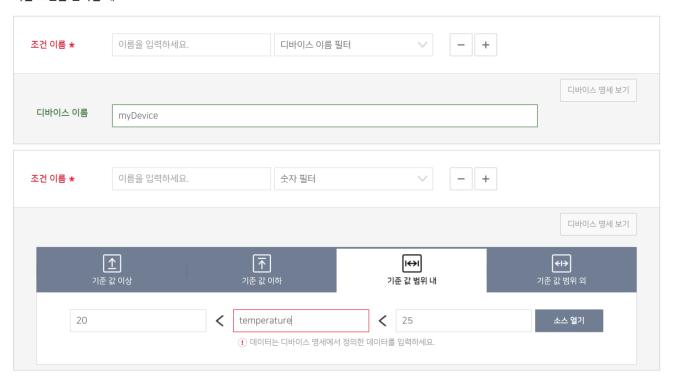
디바이스 명세 필터

서비스 내에서 명세 별로 분리하여 데이터 정제 가능

IoT 오픈 하우스

>> 룰 엔진

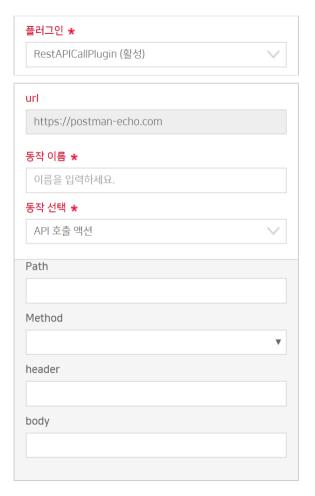
다음 조건을 만족할 때

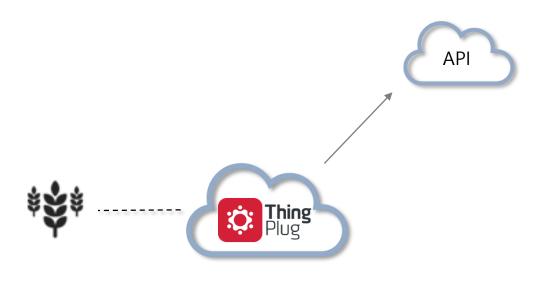


>> 룰 엔진 플러그인

Rest API Call Plugin

동작을 수행합니다



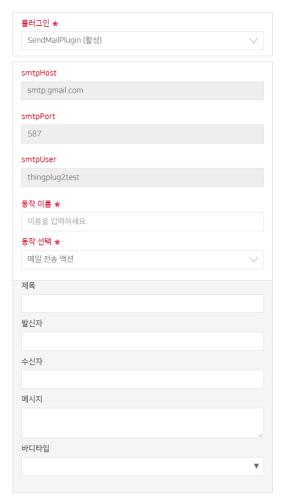


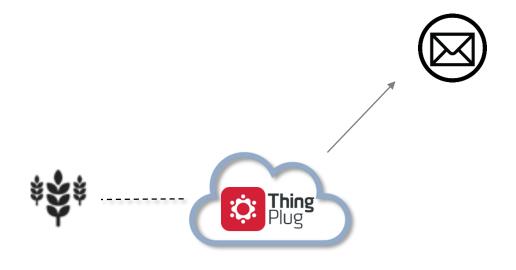
별도의 서버로 API 요청하는 플러그인

>> 룰 엔진 플러그인

Send Mail Plugin

동작을 수행합니다



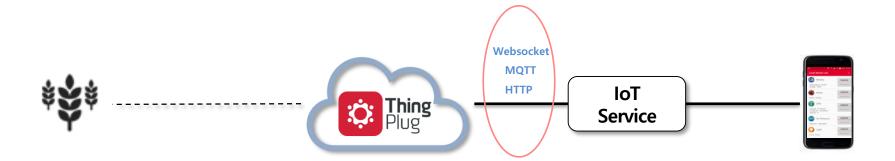


E-mail 전송

THINGPLUG API 기초

>> ThingPlug API

ThingPlug와 IoT Service 사이의 API

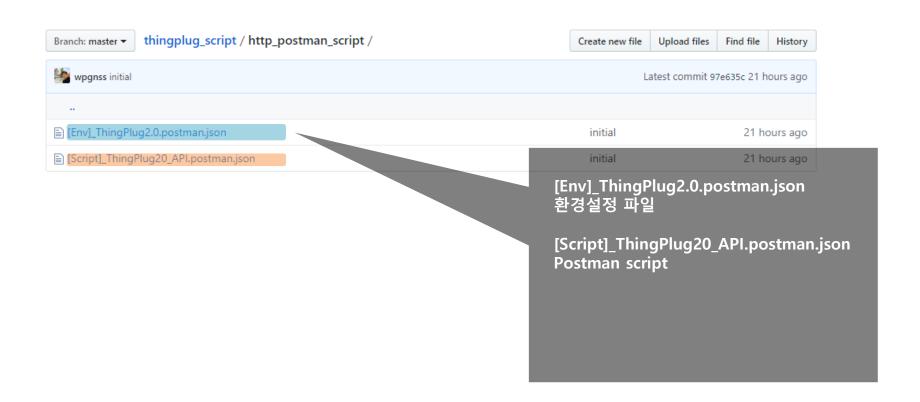


>> Postman 툴을 활용한 ThingPlug API 테스트

https://www.getpostman.com/

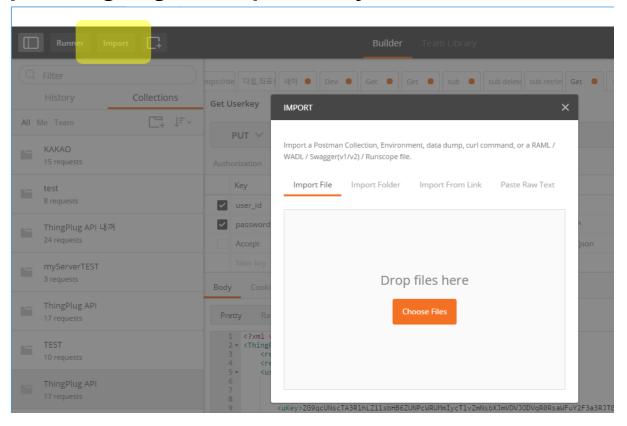


>> Postman 툴을 활용한 ThingPlug API 테스트



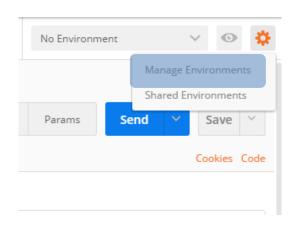
>> Postman 테스트 Script를 import 하기

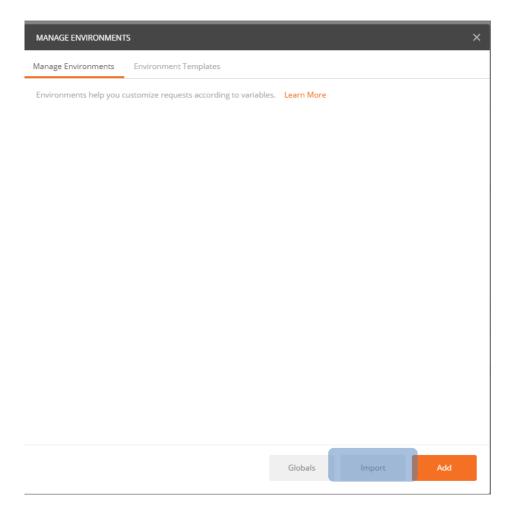
[Script]_ThingPlug20_API.postman.json 파일을 Postman에 Import 한다.



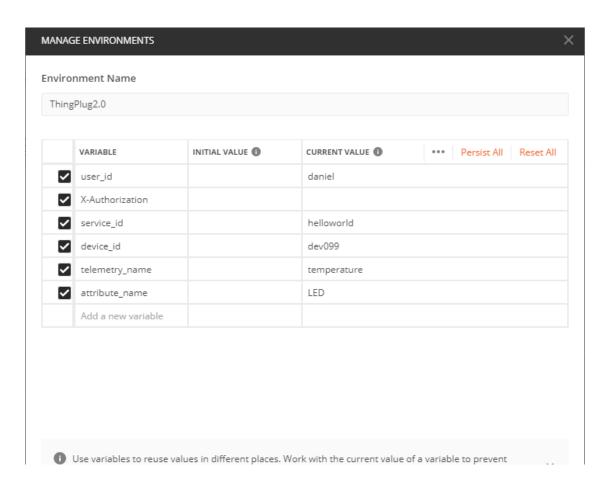
>> 환경설정 파일 import 하기

[Env]_ThingPlug2.0.postman.json 파일을 Postman에 Import 한다.





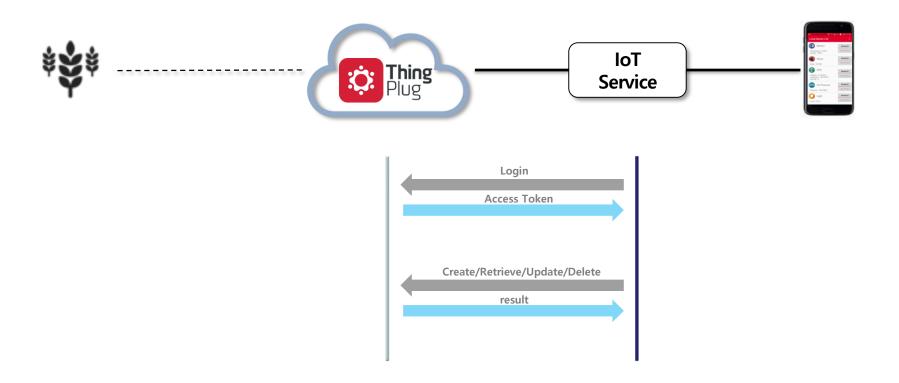
>> ThingPlug



Import 완료 후 "CURRENT VALUE"에 값을 채운다 (X-Authorization은 나중에 입력할 예정)



>> ThingPlug API 테스트 방법



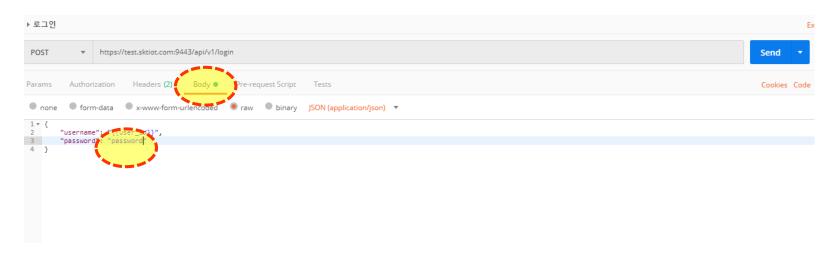
HTTP API 사용시 Access Token이 필요하다

>> accessToken 발급 받기



디바이스 관련 정보 조회

>> accessToken 발급 받기

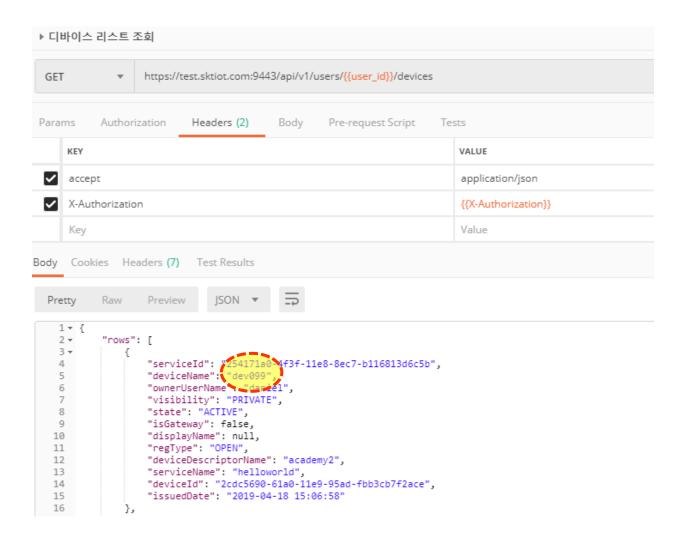


로그인 API를 사용하면 accessToken을 발급 받을 수 있다.

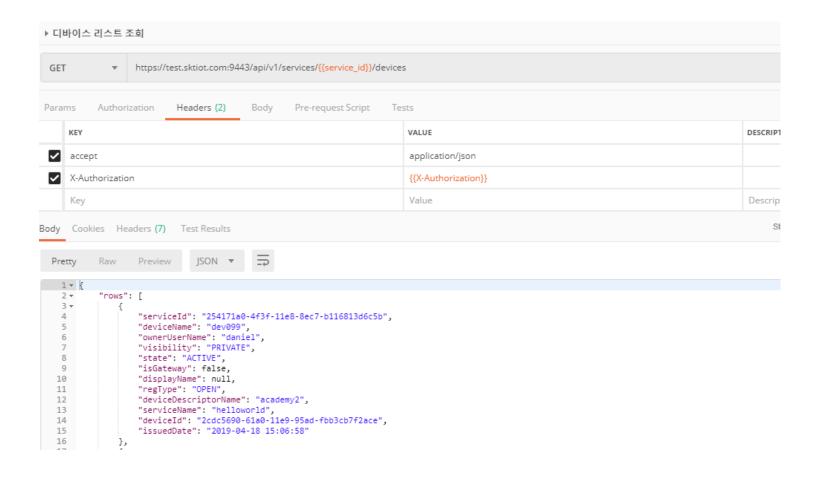
이 accessToken을 환경설정에 X-Authorization 에 입력한다.



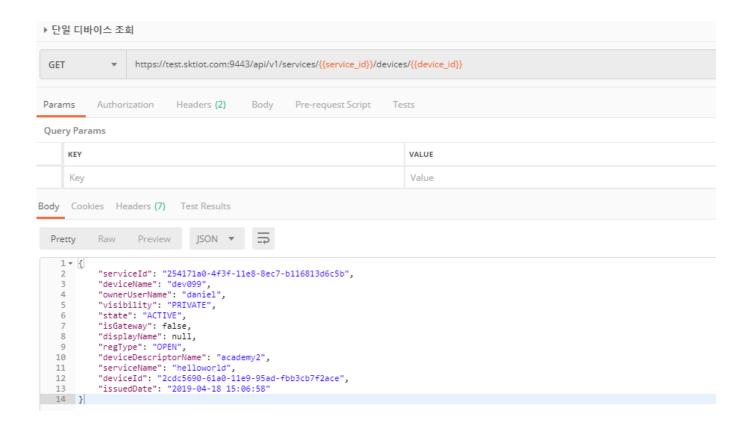
>> 내가 등록한 디바이스 리스트 조회



>> 내 서비스에 등록된 디바이스 리스트 조회

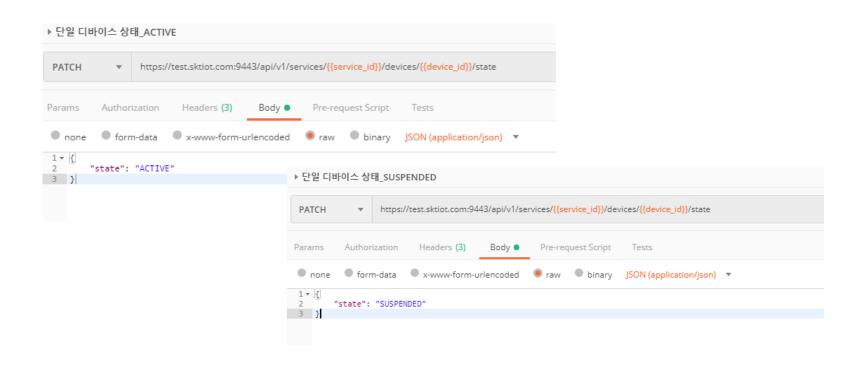


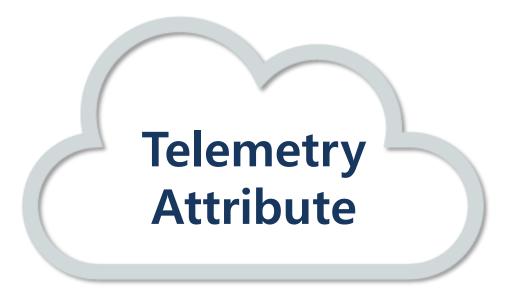
>> 단일 디바이스 조회



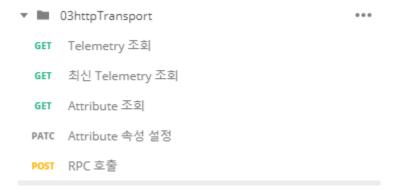
>> 디바이스 접속 제어

디바이스 active/suspend 기능을 활용하여 디바이스의 접속을 제한할 수 있다



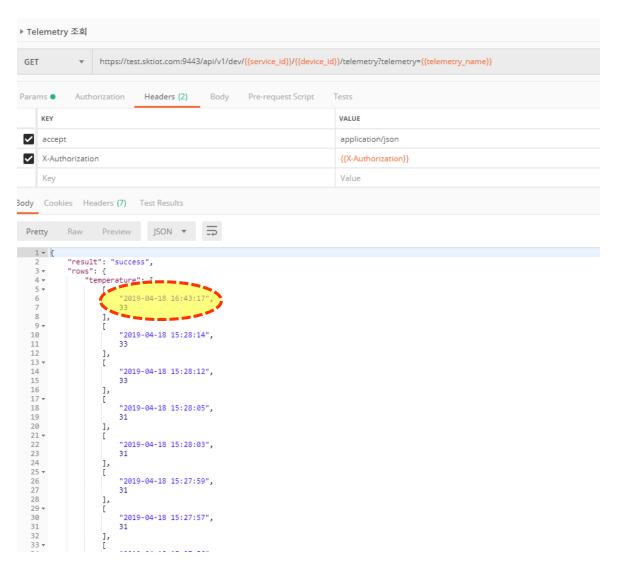


>> Telemetry/Attribute/RPC



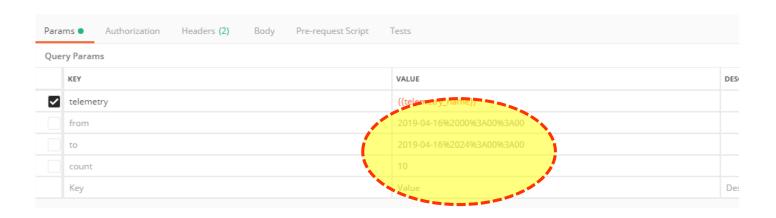
디바이스의 데이터 조회와 디바이스 제어 방법

>> Telemetry 조회



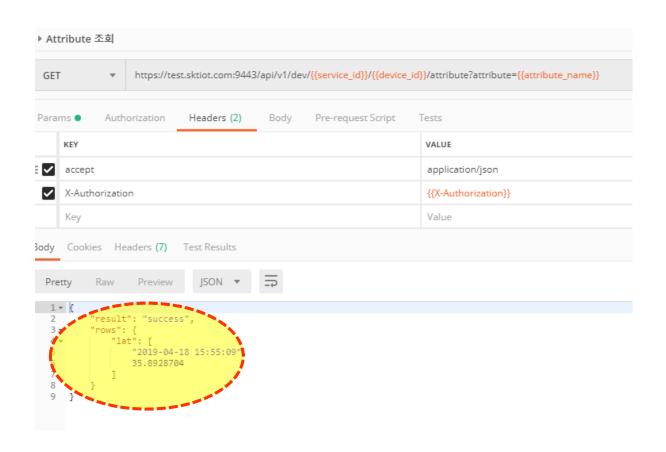
저장된 데이터를 조회

>> Telemetry 조회



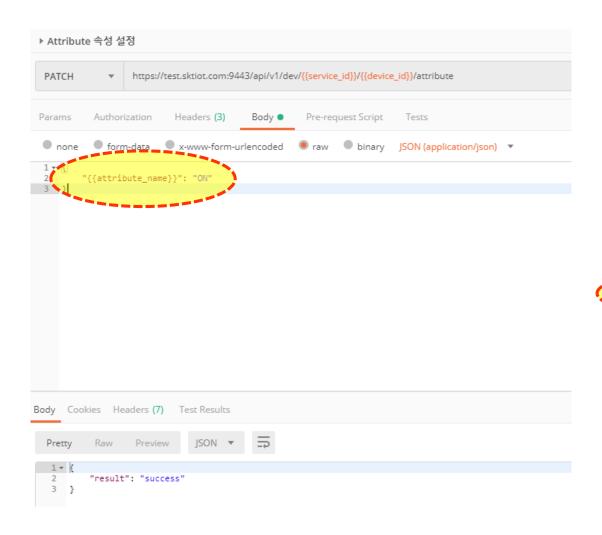
파라메터 설정에 따라 "기간", "개수" 옵션을 줄 수 있다.

>> Attribute 조회



현재 설정되어 있는 Attribute를 조회한다.

>> Attribute 속성 변경 요청

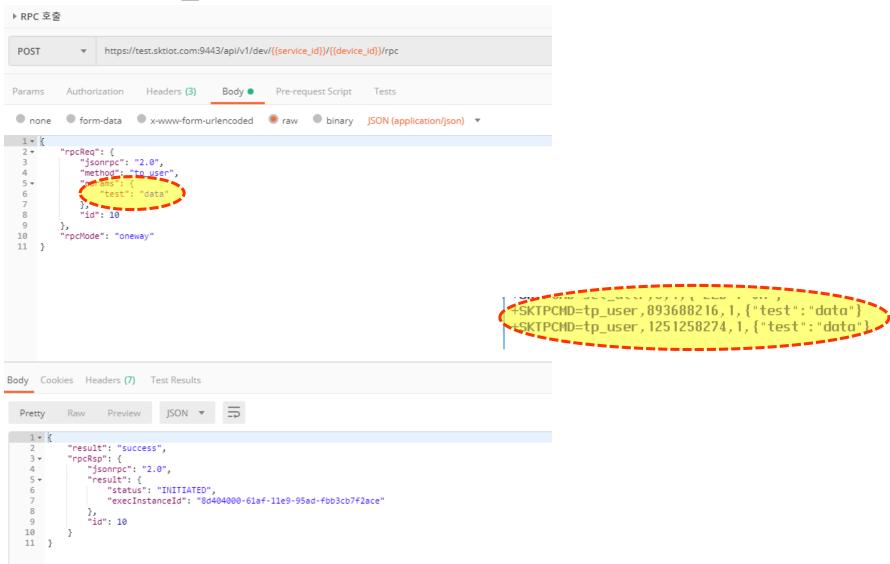


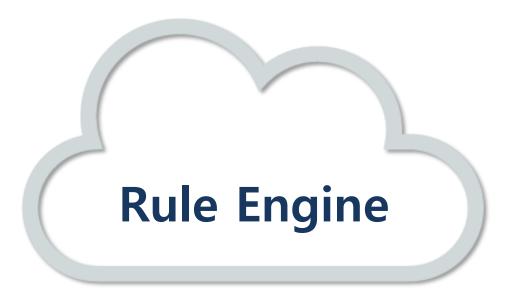
디바이스 로그에 출력된다.

```
+SKTPCMD=set_attr,0,1,{"LED":"ON"}
+SKTPCMD=set_attr,0,1,{"LED":"ON"}
+SKTPCMD=set_attr,0,1,{"LED":"OFF"}
+SKTPCMD=set_attr,0,1,{"LED":"ON"}
```

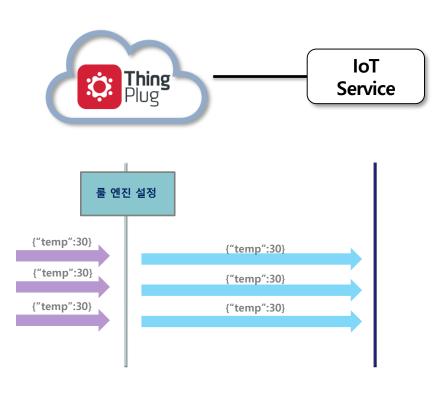
IoT 오픈 하우스

>> RPC 호출





>> 룰 엔진 – Rest API Call



서비스측 권한 획득

- ThingPlug 포탈에서 직접 룰엔진 설정

조회 가능 데이터

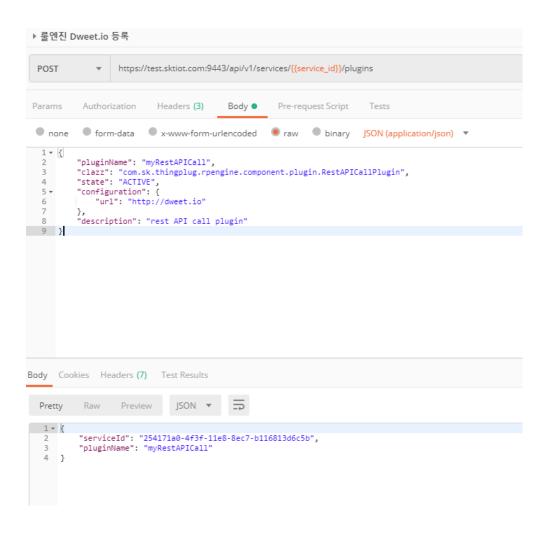
- 최신 데이터

특징

- 룰엔진 기반으로 조건/데이터 포맷 변경 가능
- 고객 서버가 Inbound port 가 오픈되어야 함

>> 룰 액션 등록 API

>> 룰 액션 등록 API



postman에서 룰 액션을 등록한다.

>> 룰 엔진 등록



기본 필터 사용, 모든 데이터에 대해 액션을 설정한다.

IoT 오픈 하우스

>> 룰 엔진 등록

플러그인 ★	myRestAPICall (활성)				
url	http://dweet.io				
동작 이름 *	post	동작 선택 *	API 호출 액션	~	
Path	/dweet/for/helloworld				
Method	POST				
header	{"Content-Type": "application/json"}				
body	{"device":"\$deviceName.id()", "temperature":\$temperature}				

Path: /dweet/for/<service_id>

Method: POST

Header: {"Content-Type": "application/json"}

Body: {"device":\$deviceName.id(), "temperature":\$temperature}

>> 룰 엔진 등록

디바이스 데이터 포함 예제

\$ + Key값의 형태로 디바이스 데이터의 value를 추출하여 body 생성(Key값은 디바이스 명세를 통해 정의 가능) 디바이스 이름을 포함하고 싶은 경우 \$deviceName.id() 포함하여 데이터 전달 가능 디바이스 데이터 중 일치하는 key가 없는 경우 \$key 형태로 데이터 전달

body	{"temperature" : \$temperature, "humidity" : \$humid, "aircon": \$airconditioner}	
body	{"device":\$deviceName.id(), "data": {"temperature" : \$temperature, "home": "address"}}	
body	습도가 \$humidity 이상입니다. 집이 너무 덥네요	

>> ThingPlug



Q&A 감사합니다