

고가람 | 안홍범 | 김규백 | 이종은 | 이상민 | 이재한 지음



ThingPlug로 시작하는 IoT 서비스개발

ThingPlug_로 시작하는 IoT 서비스개발

저자소개

고가람

SK텔레콤에서 IoT 플랫폼과 IoT 데이터 분석 관련 업무를 담당하고 있다. 서강대학교에서 학사와 석사 학위를 받았으며 삼성전자에서는 개인화 추천 플랫폼을 개발했다. 데이터 분석에 관심이 많고 새로운 기술들을 접하는 것을 좋아한다.

안홍범

SK텔레콤에서 IoT 플랫폼 개발 및 one2M, OIC 등 IoT 플랫폼 표준 업무를 담당하고 있다. one2M 표준 내 아키텍처 기술 그룹 부의장을 역임하였으며, LG전자에서는 IoT 서비스 플랫폼 프로젝트 리더로 활동하였다.

김규백

SK텔레콤에서 IoT 플랫폼 개발을 담당하고 있다. 사물인터넷포럼 기술분과위원, 경북대, KAIST에서 학부와 석사 후, 현대전자, NCSOFT, 삼성전자에서 서버 소프트웨어 개발과 데이터베이스 아키텍트 업무를, 한국오라클에서 기술 컨설턴트로 근무했다. 저서 『오라클 데이터베이스 보안』

이종은

모바일 앱과 웹 서비스를 만드는 Yo Studio와 자바스크립트 플랫폼인 타이타늄의 사용자 모임(tidev.kr)을 운영 중이다. 오픈소스 개발자이자 크리에이티브 커먼즈 활동가이며, 송실대와 카이스트를 졸업하고 다음 커뮤니케이션에서 프론트엔드를 개발했다.

이상민

SK텔레콤에서 IoT 디바이스용 플랫폼 및 소프트웨어 개발을 담당하고 있다. 지상파 DMB용 미들웨어 개발과 모바일 동영상 스트리밍용 플레이어 개발 등 모바일 디바이스용 멀티미디어 소프트웨어 개발에 매진해왔다. 최근에는 다양한 오픈소스 하드웨어에 관심을 두고 있다.

이재한

SK텔레콤에서 IoT 플랫폼을 활용하여 부산 스마트시티 플랫폼을 만들고 있다. 서울대에서 학사와 석사를, 텍사스 A&M 대학교에서 무선 센서 네트워크로 박사 학위를 받았다. LG전자와 삼성SDS 연구소를 거쳤다.



들어가며

‘아, 저거 나도 생각했던 건데.’

기발한 아이디어로 성공한 서비스를 보면서 한 번쯤 이런 생각을 해본 적이 있을 것입니다. 이런 것이 있으면 좋겠다고 생각만 하고 넘어간 것들이 실제로 만들어지고 성공하는 모습을 보면 가끔 배가 아프기도 합니다. 하지만 대부분의 경우 기술이나 시간, 비용 문제가 떠오르면서 어차피 직접 만들긴 어려웠겠다는 생각에 미련을 접고 일상으로 돌아갑니다.

스마트폰이 생기고 다양한 서비스와 앱들이 등장하면서 이런 경험을 좀 더 자주 하게 됩니다. 그리고 놓친 아이디어에 대한 아쉬움을 더 커지곤 합니다. 예전에는 ‘어차피 만들긴 힘들었을거야’라고 위안할 수 있었지만, 이젠 달라졌기 때문입니다. OS나 웹을 통해 제공되는 다양한 API들을 활용하면 손쉽게 앱을 만들 수 있고 앱스토어를 통해 배포도 가능합니다. 백엔드 서버가 필요한 서비스라면 클라우드를 활용해서 적은 비용으로 만들 수 있습니다. 아이디어가 있으면 큰 어려움 없이 생각하고 있는 것을 만들어 볼 수 있는 세상이 된 것입

니다. 이러한 세상을 가능하게 한 가장 큰 요인이 바로 플랫폼입니다.

이미 우리는 다양한 종류의 플랫폼을 잘 이용하고 있습니다. 사람들이 유용하게 쓰고 있는 다양한 스마트폰용 앱들은 안드로이드나 iOS를 플랫폼으로 사용합니다. 마우스 클릭 몇 번이면 클라우드 환경을 구성할 수 있는 아마존의 플랫폼도 있습니다. 페이스북이나 트위터는 이미 단순한 소셜네트워크 서비스를 넘어 오픈 API를 기반으로 다른 서비스들을 재생산하는 플랫폼으로 진화했습니다. 쉽게 모바일 앱을 만들 수 있도록 유용한 API는 물론 BaaS(Backend as a Service) 같은 클라우드 서비스를 제공하는 플랫폼도 있습니다.

IoT에서도 IoT 서비스를 만들고 운영하는 것을 지원하기 위한 플랫폼이 등장하기 시작했습니다. ThingPlug는 SK텔레콤이 만든 대표적인 IoT 플랫폼입니다. SK텔레콤은 이동통신사업자로서 M2M 플랫폼을 운영하면서 다양한 M2M 기기와 서비스를 지원해 왔습니다. ThingPlug는 이 다년간에 걸친 M2M 플랫폼 운영 경험과 노하우가 바탕이 되어 만들어졌습니다. 따라서 M2M 플랫폼의 장점을 그대로 이어받으면서 IoT 시대에 걸맞은 다양한 서비스 기능들을 갖추고 있습니다.

이 책에서는 독자에게 이러한 ThingPlug의 다양한 기능들을 소개하고 직접 사용해 볼 수 있는 방법들을 설명하고 있습니다. 책을 읽고 독자가 IoT나 ThingPlug에 대한 새로운 지식을 얻는다면, 그리고 실제로 플랫폼을 활용해 아이디어를 구현하는 데 도움을 받을 수 있다

면 좋겠습니다. 마치 플랫폼 위에서 IoT 서비스가 구체화되고 새로운 가치가 창출되는 것처럼 이 책도 IoT 서비스를 만들려고 하는 사람들에게 그러한 발판이 될 수 있기를 바랍니다.

이 책은 특정 사전 지식을 갖고 있는 독자만을 대상으로 하지는 않습니다. IoT에 관심은 있지만 어디서부터 접근해야 할지 몰라 망설이던 개인 개발자부터, IoT 서비스를 만드는데 구체적인 플랫폼의 도움이 필요한 사람까지 넓은 범위의 독자가 나름대로 의미있는 것을 얻을 수 있도록 구성했습니다. IoT나 IoT 플랫폼인 ThingPlug가 생소한 독자라면 처음부터 찬찬히 읽는 것을 추천하고, 어느 정도 사전 지식이 있는 독자라면 필요한 기능과 관련된 부분만 취사선택해도 좋습니다. 어떤 배경 지식을 가지고 있는 독자건 나름의 새로운 가치를 얻을 수 있기를 바랍니다.

ThingPlug는 빠르게 발전하고 진화하는 플랫폼입니다. 이 책을 통해 ThingPlug에 대해 기본적인 것을 익히셨다면 홈페이지를 방문해서 매일 진화하고 있는 ThingPlug를 직접 사용해 보시기 바랍니다. ThingPlug는 IoT 서비스에 관심이 있는 누구에게나 열려 있습니다.

IoT 서비스를 시작하는 첫걸음을 ThingPlug와 함께 한다면 여러분은 마음 편히 서비스의 핵심에 집중할 수 있을 것입니다. 플랫폼의 힘이 IoT에도 찾아오고 있습니다.

차례

저자소개	5
들어가며	7

CHAPTER 1 IoT 플랫폼, ThingPlug의 등장

15

왜 IoT인가?

17

사물이 인터넷을 만날 때

17

IoT 플랫폼이라는 필연

21

ThingPlug란?

26

ThingPlug를 만나는 세 가지 방법

26

ThingPlug의 장점

39

CHAPTER 2 oneM2M 표준을 탑재한 ThingPlug

43

oneM2M 기반 플랫폼 ThingPlug

45

IoT 서비스를 위해 등장한 플랫폼 표준, oneM2M

45

oneM2M 표준을 수용한 ThingPlug

48

ThingPlug에 탑재된 oneM2M

51

oneM2M의 주요 구성 요소

53

oneM2M CSE가 제공하는 공통 기능

55

ThingPlug에 구현된 REST API의 기본 구조와 호출하는 방법	56
ThingPlug의 oneM2M 인터페이스 활용하기	62
ThingPlug에 oneM2M 장치 등록하기	62
ThingPlug에 oneM2M 인터페이스로 데이터 저장하기	65
ThingPlug로 관심 데이터 변경을 모니터링하는 방법	70
장치의 위치를 쉽게 획득하고 관리하는 방법	73
ThingPlug 플랫폼을 통해서 장치 제어 API를 호출하는 방법	77
oneM2M API의 통신 프로토콜	79
IoT의 뜨거운 감자 ‘보안’	84
ThingPlug에서 제공하는 oneM2M 접근 제어 기능	86

CHAPTER 3 oneM2M을 직접 만져 보자! 89

ThingPlug 스타터킷으로 oneM2M 시작하기	91
스타터킷의 구성 요소	91
사전 준비	93
스타터킷 다운로드	95
config.js 파일 생성 및 설정	97
IoT 장치의 실행	99
장치 등록	100
장치 데이터 전송	106
애플리케이션 실행	108
ThingPlug 서버에 저장된 최신 데이터 조회	110
애플리케이션이 보내는 장치 제어명령	111
스타터킷을 활용한 간단한 서비스 만들기	119
예제 시나리오: 웹 서버로서의 애플리케이션	119

CHAPTER 4 나를 위한 맞춤형 GMMP 127

GMMP 기능 알아보기 129

GMMP 기본 기능 131

GMMP 확장 기능 136

예제를 통해 GMMP 살펴보기 140

사전 준비하기 140

디바이스 인증 144

디바이스 데이터 전송 150

디바이스 제어 명령어 처리 154

애플리케이션 개발하기 158

디바이스 전송 데이터 확인하기 163

디바이스 제어 명령어 보내기 165

CHAPTER 5 코딩할 줄 몰라도 가능한 IoT? 167

ThingPlug 서비스 플랫폼 169

ThingPlug 서비스 플랫폼의 특징 및 기능 175

SaaS 클라우드 서비스 175

위젯 기반의 반응형 웹 UI 제공 176

다양한 기능의 위젯으로 꾸미는 대시보드 177

규칙 기반의 이벤트 처리 180

통계 및 센서 데이터 분석 기능 183

타임라인과 태그 185

ThingPlug 서비스 플랫폼 연동방법 186

환경 설정	187
비글본 블랙 설치	189
비글본 블랙 등록	190
게이트웨이 모델 선택과 대시보드 설정	194
ThingPlug 서비스 플랫폼으로 당장 이렇게 만들 수 있다	198
버섯 농장 모니터링 서비스	198
오피스 건물 모니터링 서비스	202

CHAPTER 6 ThingPlug 이모저모 207

ThingPlug와 오픈소스 하드웨어	209
스마트 온도조명	214
스마트 “빨래 끝” 알리미	220
스마트 화분	221
스마트 환기 알리미	222
ThingPlug와 빅데이터 분석	224
샘플링과 빅데이터	224
IoT와 빅데이터, 그리고 ThingPlug	226
ThingPlug에서의 데이터 분석	231

CHAPTER 7 지금 만나 보는 ThingPlug 241

ThingPlug 이렇게 활용된다	243
커넥티드 카	244
전자발찌	248
원격 검침	250

스마트 팜과 스마트 양식장	251
스마트 홈	253
스마트 시티 플랫폼에서 활용되는 ThingPlug	255
스마트 파킹 서비스	256
스마트 가로등	258
스마트 미아 방지 서비스	259
엄마·아빠들, 수유 걱정 덜어드려요 —‘맘마미아’팀을 만나다	263
마치며	267

CHAPTER

1

IoT 플랫폼, ThingPlug의 등장



왜 IoT인가?
ThingPlug란?
ThingPlug의 장점



재미있고 유용한 IoT 서비스들이 등장하고 있습니다. 스마트 홈과 같은 우리 삶과 밀접한 영역에서부터 Industry 4.0처럼 제조 공정 같은 전문적인 영역까지 다양한 분야에서 IoT 서비스를 통해 새로운 가치가 만들어지고 있습니다. IoT는 이제는 전문적인 소수만 접근할 수 있는 영역이 아닙니다. 마치 스마트폰 앱처럼 누구나 아이디어만 있으면 재미있는 IoT 서비스를 만들 수 있습니다. 이를 위한 IoT 플랫폼이 있기 때문입니다. 개인 개발자의 DIY 서비스부터 대규모 IoT 서비스까지 모두 IoT 플랫폼을 통해 쉽게 'IoT화' 할 수 있습니다.

ThingPlug는 IoT 서비스를 좀 더 쉽고 빠르게 만들고 안정적으로 운영할 수 있도록 지원하는 다양한 기능을 갖추고 있는 SK텔레콤의 IoT 플랫폼입니다. IoT 서비스를 만들고자 하는 모든 이들에게 공개되어 있어 누구나 사용할 수 있습니다. 이번 장에서는 IoT 서비스를 만드는 데 필요한 요소들과 IoT 플랫폼을 활용해 얻을 수 있는 이점이 무엇인지, 그리고 IoT 플랫폼으로서 ThingPlug가 제공하는 기능과 ThingPlug를 사용하기 위해서는 어떻게 해야 하는지에 대해 간략히 살펴보도록 하겠습니다.

왜 IoT인가?

사물이 인터넷을 만날 때

IoT^{Internet of Things}, 사물 인터넷이 인기입니다.

우리가 익히 알고 있는 그 인터넷에 기계를 포함한 다양한 것^{Thing}들이 연결된다는 뜻입니다. IoT라는 용어가 등장하기 전에도 유비쿼터스, M2M^{Machine to Machine}처럼 기계나 사물끼리 정보를 주고받고 통신하는 것을 가리키는 용어가 있었습니다. 하지만 M2M은 단순히 기계끼리 연결된다는 것이고, 유비쿼터스는 실체가 모호한 개념이었습니다. 이 둘보다 확장되고 명확한 개념인 IoT는 어떤 기술을 기반으로 하고 있고 어떤 특징이 있을까요?

중요한 계기 중 하나는 ‘센서’의 발달입니다. 사람이 감각을 통해 현실을 받아들이듯이 사물은 센서를 통해 현실 정보를 인식할 수 있습니다. 센서 개발에 필요한 반도체 소자 설계 기술이 발전하면서, 다양한 종류의 센서가 저렴한 가격으로 만들어지기 시작했고, 사물을 통해 수집할 수 있는 정보의 종류가 다양해졌습니다. 이렇게 사물이 센서를 통해 감각을 갖게 된 것에서부터 IoT가 시작됐다고 해도 과언이 아닙니다. 교환할 수 있는 정보가 생기면서 IoT에서 가장 중요한 ‘연결’이 의미를 가지기 때문입니다.

또 다른 계기는 통신 기술의 발달입니다. 기존에는 특별한 기계나 장치만이 인터넷에 연결될 수 있었지만, 통신 칩이 소형화되고 강력해져 사물 안으로 들어갈 수 있게 되면서 연결된 사물의 종류도 폭발적으로 늘어나게 됐습니다. 존재하는 거의 모든 것이 IoT의 '사물 Thing'로서 연결될 수 있는 시대가 된 것입니다.

IoT에서 말하는 '사물'은 연결을 전제로 하고 있습니다. 아무리 많은 센서를 가진 사물이라도 다른 사물이나 사람에게 정보를 전달하지 못하면 의미가 없습니다. 오히려 하나의 감각, 한 가지 종류의 센서만

BOX

M2M이란?

M2M은 말 그대로 기계 간 통신을 뜻합니다. M2M이란 유통에서 자원, 교통에 이르기까지 다양한 산업 분야의 특화된 기기류를 연동할 수 있게 하는 기술로 오랜 기간 사용되어 온 것이기도 합니다. 기기류를 사람의 개입 없이도 서로 연결하는 기술이라는 점은 IoT와 같지만 M2M과 IoT는 '연결' 자체에 집중하느냐와 연결을 통해 만들어질 수 있는 '서비스에 집중하느냐' 하는 차이가 있습니다.

예를 들어 ITS(지능형 교통망, Intelligent Transport System)은 예전부터 교통 분야에서 각종 센서 및 CCTV 등을 전산에 연결하고 자동화 하기 위해 꽤 각광 받는 분야였습니다. M2M 시절에는 센서에서 제어장치, 카메라까지 다양한 기기들을 연결하기 위한 프로토콜과 연결 기술이 추가 되었습니다. 하이패스 단말기처럼 DSRC(Dedicated Short Range Communication)이라는 근거리 전용 통신 기술이나 PLC(Power Line Communication)같은 전력선 통신 기술이 개발되기도 했습니다. IoT 시대가 되면서 이런 기반 기술 위에 구성되는 서비스가 추가 되고 있습니다. 교통 정보를 활용한 신호등 시스템 개선부터 자율 주행 자동차까지 '연결'에 기반한 '서비스'가 만들어지고 있는 것입니다.

가진 사물이라도 인터넷에 연결되고 정보를 전달할 수 있다면 거기
에서부터 새로운 의미가 만들어질 수 있습니다. 다양한 사물들이 수
집한 정보가 인터넷을 통해 클라우드로 모이고 이중의 정보들이 엮
이면서 그 안에서 새로운 가치가 탄생하게 되는 것입니다. 마치 원격
통신을 위해 만들어진 인터넷이 발판이 되어 초기에는 상상하지 못
했던 서비스들이 만들어지고 엄청난 기회가 생겨난 것과 같습니다.

연결은 단지 눈에 보이는 사물에만 국한된 것이 아닙니다. 소프트웨
어적인 객체와 같이 컴퓨터 안에만 존재하는 개념이나 이벤트, 또는
서비스들도 IoT의 일부가 되어 서로 소통할 수 있습니다. 그리고 이
미 인터넷의 중심이던 사람과도 연결이 됩니다. 사람만이 사용하던
인터넷이 사물이라는 새로운 대상과 만나게 되면서 IoT의 시대가 펼
쳐지게 된 것입니다.

이처럼 IoT는 센서 기술부터 통신 기술의 발달, 클라우드를 비롯한
인터넷의 발전이 서로 융합되면서 만들어진 것입니다. 어쩌면 이런 기
술들이 충분히 성숙한 지금 IoT가 주목받는 것은 당연할 수도 있습
니다. IoT는 그 이전의 M2M이 그랬던 것처럼 하나의 응용 분야에
특화된 기술이 아닙니다. 마치 인터넷처럼 모든 분야에 활용될 수 있
는 기술입니다. 게다가 사물에서 시작하기 때문에 오히려 인터넷보다
더 넓게 오프라인에서 온라인까지 모두를 아우를 수 있습니다. O2O
등 생활밀착형 IT에 대한 관심이 커지고 있는 지금 오프라인과 온라

인의 융합을 통해 우리 삶의 혁신을 가져오는데 IoT가 큰 역할을 할 수 있을 것입니다.

다양한 종류의 사물들이 연결되면서 IoT 비즈니스의 기회는 점점 확대되고 있습니다. 그 결과로 헬스케어와 같은 개인 건강 관리, 일반 가정집부터 빌딩까지 포함한 시설관리, 농업, 임업, 양식업 같은 1차 산업, 공공 안전이나 서비스 등까지 새로운 시도들이 계속되고 있습니다.

이제 이 가능성을 현실화하고 IoT 서비스로 구체화하는 것은 우리들의 몫입니다. 그리고 그런 우리에게 IoT 플랫폼 ThingPlug가 든든한 지원군이 되어줄 것입니다.

BOX

IoT와 유비쿼터스

20세기가 끝나고 21세기가 시작될 무렵 유비쿼터스라는 말이 유행이었습니다. 이 단어가 이미 10여 년 전부터 유행했음을 생각해 보면, 세상은 빨리 바뀌는 듯하면서도 의외로 면면히 바뀌어 갑니다. U-시티 등 온갖 '유(U)'로 시작하는 신조어가 난무하던 시절이었습니다. 이미 그 시절부터 언제 어디서나 인터넷에 연결되는 세상, 모든 것이 이미 연결된 세상에 대한 미래상이 그려지고 있었던 것입니다.

그런데 스마트폰 혁명 이전이다 보니, 기간 기술은 어느 정도 정비되어도 가장 말단을 위한 세세한 기술까지는 최적화가 되어 있지 않았습니다. 게다가 아직 전자 기술의 집적도도 요즘 같지 않아서 간단한 일을 위한 기판도 작지 않은 크기이다 보니, 사물 안에 쑥 들어가서 몰래 숨는 일도 쉽지만은 않았습니다. 그래서일까요? 많은 세월이 흐른 지금 이 분야에 대해 약간의 의구심을 가지고 보는 시각도 당연히 생겨날 수밖에 없었고, IoT를 트러블 인터넷(Internet of

Trouble), 그러니까 문제를 일으키는 인터넷이라고 부르는 일까지 생기게 되었습니다.

하지만 이와 같은 답답함의 추억 덕에 오늘날의 IoT 플랫폼이 생겨났다고 해도 과언이 아닙니다. 그때의 시행착오 하나하나가 진정한 사물인터넷이 개화하도록 한 원동력인 셈입니다.

IoT 플랫폼이라는 필연

그렇다면 IoT에서 플랫폼은 어떤 역할을 하는 것일까요? 플랫폼이 없더라도 사물들이 인터넷에 연결되고 정보를 어딘가로 전달하는 데에는 문제가 없을 것입니다. IoT 서비스를 만드는 데 필요한 모든 것을 플랫폼 없이 직접 구성하는 것도 불가능한 일은 아닙니다. 하지만 IoT 플랫폼을 이용해 IoT 서비스를 구성하게 되면 많은 장점이 있습니다. 어떤 장점이 있는지 알아보기 위해 먼저 IoT 서비스를 제공하는 데 필요한 구성요소가 어떤 것이 있는지 살펴보겠습니다.

» IoT 서비스의 구성 요소

아무래도 IoT가 사물들의 인터넷인 만큼 구성 요소 중 첫 번째로 생각할 수 있는 것은 바로 ‘사물’ 그 자체일 것입니다. IoT 서비스는 기존에 존재하지 않던 새로운 사물을 만드는 것은 물론 사용자에게 새로운 가치를 줄 수 있도록 기존의 사물을 개선하는 것을 목표로 하는 경우가 많습니다. 이때 가장 중요한 역할을 하는 것이 센서입니다.

IoT에서 센서는 따로 떼어 생각할 수 없을 만큼 중요한 역할을 담당합니다. 사람에게 주변 환경을 인식하는 감각이 있는 것처럼, 센서는 사물에 감각을 부여하여 다양한 정보를 수집할 수 있게 합니다. 따라서 IoT 시대의 사물들은 센서를 통해 서비스에 필요한 감각을 갖게 된 새로운 종류의 사물이라고 볼 수 있습니다. 그리고 액추에이터를 부착하여 필요한 경우 작동할 수 있게 하거나 LED, 화면 등을 부착하여 외부에서 전달받은 정보를 사람에게 전달하는 등의 다양한 기능을 제공할 수도 있습니다.

이처럼 기존의 사물과 달리 감각을 갖고 작동할 수 있거나 정보를 전달할 수 있는 기능을 갖춘 사물이 준비됐다면 이제 이 사물들을 연결할 차례입니다.

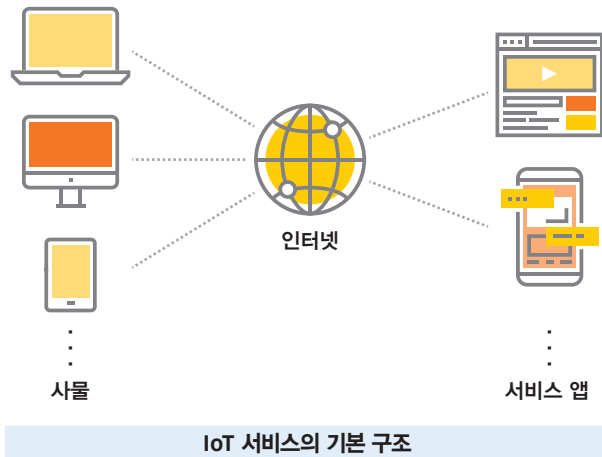
서비스의 두 번째 구성 요소는 ‘연결’입니다.

하나의 사물로는 할 수 없던 일들이 서로서로 이어지고 또한 인터넷에 연결됨으로 인해서 가능해지는 것이 IoT 서비스입니다. 사물의 정보를 인터넷을 통해 언제 어디서든지 확인할 수 있고, 원격으로도 바로 옆에 있는 것처럼 제어할 수 있는 ‘스마트 썩Smart Thing’이 될 수 있게 하는 기본. 그것이 바로 연결입니다. IoT 서비스가 기존의 제품들과 차별화된 가치를 주는 이유는 사물이 어딘가에 연결되어 있기 때문이라고 해도 될 것입니다.

세 번째 구성요소는 ‘서비스 앱’입니다. 사물과 사물 간의 연결에서는

필요치 않을 수도 있지만, 사물과 사람의 연결에서는 접점이 되어주고 사람이 이해하기 쉽게 정보를 보여줄 서비스 앱이 꼭 필요합니다. 서비스 앱은 안드로이드나 iOS 등의 앱으로 만들어질 수도 있고 브라우저를 통해 접근하는 웹서비스 형태로 만들어질 수도 있습니다. 하지만 어떤 형태로 되어 있건 서비스 앱은 사물의 정보를 확인하고 제어할 수 있는 기능을 제공하고 사용자에게 새로운 가치를 전달하는 것을 목적으로 하고 있습니다.

IoT 서비스의 세 가지 구성요소를 그림으로 표현하면 다음 그림과 같습니다. 사물들과 서비스 앱이 존재하고 그것들이 서로 연결되어 우리가 지금 사용하는 인터넷처럼 정보를 주고받을 수 있도록 구성되는 것이 IoT 서비스의 기본 구조입니다.



하나의 IoT 서비스가 완결성을 갖고 서비스되기 위해서는 적어도 이 세 가지가 모두 구성되어야 합니다. 하지만 백지 상태에서 이 모든 구성 요소를 개발하고 또 운영까지 하는 것은 쉬운 일이 아닙니다.

사물의 하드웨어를 만들고 그 안에서 동작하는 소프트웨어를 개발하는 것과 사용자에게 선보일 서비스 앱의 UI를 개발하는 것은 각기 다른 전문 영역의 일입니다. 이들 사이의 연결을 잘 관리하고 적절한 타이밍에 데이터를 주고받으면서 서로 대화할 수 있게 하는 것 역시 만만치 않은 작업입니다. 무엇보다도 복잡하고 힘든 작업은 사물과 서비스 앱이 항상 잘 연결되어 있도록 서비스를 운영하는 것입니다. 이러한 서비스 운영을 위해서는 사물이 항상 연결되어 있도록 연결성 **Connectivity**을 확보한 후 이를 다시 잘 유지해야 합니다. 또 연결된 사물이 확산될수록 늘어나야 하는 대규모 서버에 관한 기술 역시 필요합니다. 이 모든 기술은 어느 하나 만만하지 않은 전문 영역입니다. 따라서 IoT 서비스를 밑바닥부터 만든다면 규모 있는 개발팀이 있어야 할 것입니다.

하지만 IoT 플랫폼을 활용한다면 이런 부분을 해결할 수 있습니다. IoT 플랫폼에서는 IoT 서비스를 만들 때 개발이나 운영에 꼭 필요한 공통적인 부분을 대신 제공해줍니다. 이런 기능들을 활용하면 서비스를 훨씬 빠르고 쉽게 만들 수 있고 안정적으로 운영할 수도 있습니다.

대부분의 IoT 플랫폼은 가장 기본적이면서도 까다로운 구성 요소인 ‘연결’에 대한 부분을 책임지고 제공합니다. 따라서 사용자는 이 부분에 대한 고민을 덜고 서비스의 핵심을 잘 만드는데 집중할 수 있습니다. 그리고 또 다른 구성 요소인 ‘사물’이나 ‘서비스 앱’ 개발을 지원하기 위해서도 API나 라이브러리^{Library}, SDK^{Software Development Kit} 등을 제공하기 때문에 역시 훨씬 효율적으로 IoT 서비스를 개발할 수 있습니다.

플랫폼을 활용하면 개발 측면에서만 이점을 얻을 수 있는 것이 아닙니다. 플랫폼에서 사물과 서비스 앱의 연결을 관리하고 상태를 모니터링할 수 있도록 도와주기 때문에 훨씬 더 안정적으로 서비스를 운영할 수 있습니다. 서비스의 사용자가 갑자기 증가하거나 하드웨어가 고장나는 것에 대해서도 걱정없이 무중단 서비스가 가능합니다. 이처럼 여러 장점이 있기 때문에 대부분의 IoT 서비스들이 플랫폼을 활용해서 개발되는 것은 어찌보면 당연하다고 볼 수 있습니다.

이 책에서 소개하고 있는 SK텔레콤의 IoT 플랫폼, ThingPlug도 이런 여러가지 플랫폼의 기본 기능들을 사용자에게 제공합니다. 그리고 다른 IoT 플랫폼들과 차별화되는 특별한 기능들 역시 제공하고 있습니다. 그러면 ThingPlug의 기능과 장점에 대해 살펴보도록 하겠습니다.

ThingPlug란?

ThingPlug라는 이름은 IoT^{Internet of Things}의 사물을 의미하는 ‘Thing’과 전기 콘센트에 끼우는 플러그의 ‘Plug’가 합쳐진 합성어로 누구나 아이디어만 있으면 IoT 서비스를 만들 수 있다는 뜻입니다. IoT의 아이디어가 실현될 수 있도록 지원하고자 하는 SK텔레콤의 큰 의지가 담긴 플랫폼입니다.

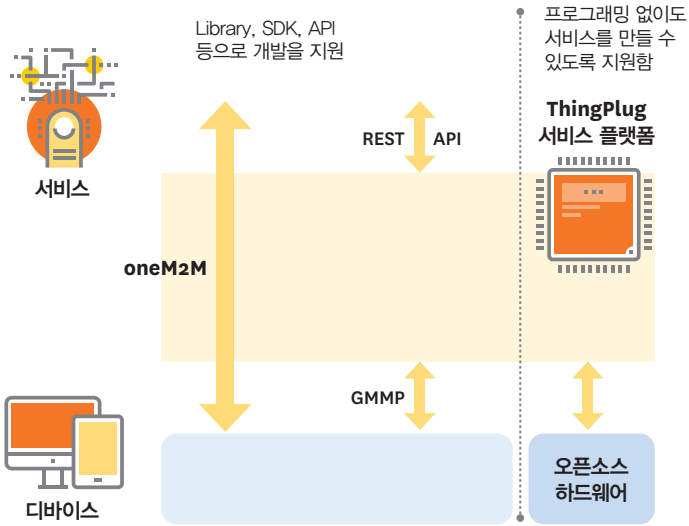
전기 제품은 언제 어디서나 콘센트에 플러그를 끼우는 것만으로도 사용할 수 있습니다. 마찬가지로 IoT의 사물들도 ThingPlug에 연결하면 언제 어디서나 IoT의 사물로서 기능할 수 있습니다. 그리고 이러한 연결이 플러그를 전기 콘센트에 끼우는 것만큼 간단하게 이루어질 수 있도록 발전시키는 것이 ThingPlug의 목표입니다.

ThingPlug에서 제공하는 다양한 기능들은 앞으로 하나씩 설명하도록 하고, 여기서는 ThingPlug를 활용하기 위한 세 가지 방식에 대해 설명하겠습니다.

ThingPlug를 만나는 세 가지 방법

IoT 서비스는 그 규모나 서비스 목적 등에 따라 필요한 기능이 다릅니다. ThingPlug에서는 다양한 요구 사항을 갖고 있는 IoT 서비스들

을 잘 지원하기 위해서 선호에 따라 선택할 수 있는 세 가지 방식을 지원하고 있습니다. 다음 그림은 ThingPlug를 활용해서 IoT 서비스를 만드는 세 가지 방식에 대해 보여줍니다.



» 국제 표준 oneM2M

첫 번째 방식은 국제 표준인 oneM2M을 이용해서 IoT 서비스를 개발하는 것입니다.

IoT 플랫폼과 디바이스 혹은 서비스가 정보를 주고받기 위해서는 정해진 프로토콜에 따라 연동되어야 합니다. 프로토콜은 크게 누구나 사용할 수 있게 공개된 프로토콜과 특정 단체나 회사에서만 폐쇄적으로 사용하는 프로프라이어터리 **Proprietary** 프로토콜이 있습니다. 사

유, 전용, 독자적 프로토콜이라고도 불립니다.

일반적으로 ETSI나 ISO 같은 국제 표준화 단체에서 정의하고 전 세계적으로 사용하기 적합한 표준을 만들어서 공개하는 것을 국제 표준이라고 하는데, IoT 영역에서는 oneM2M이 가장 대표적인 국제 표준입니다. ThingPlug는 바로 이 oneM2M의 호환 플랫폼입니다.

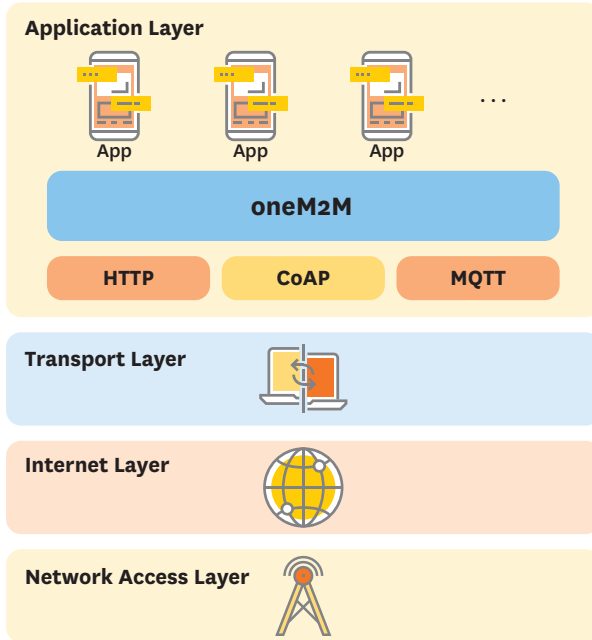
국제 표준을 준수하는 플랫폼을 사용하게 되면 프로프라이에터리 프로토콜을 지원하는 플랫폼을 사용하는 것에 비하여 다양한 장점을 얻을 수 있습니다. 웹 표준으로 유명한 HTTP가 대표적인 국제 표준의 예입니다. 다양한 브라우저와 웹에 연결된 모든 웹서버들이 국제 표준인 HTTP를 사용하기 때문에 사용자는 선호하는 브라우저를 사용할 수 있고 웹서버 역시 접속하는 클라이언트별로 다른 프로토콜을 지원하느라 고생하지 않아도 웹서비스를 할 수 있습니다.

이는 마치 윈도우 전용 프로그램을 만들면 맥이나 리눅스에서 작동하지 않지만 웹으로 만들면, 특히 국제표준인 HTML5로 프로그램을 만들면 어디에서나 작동하고 서로 연동 가능해지는 것과 비슷합니다. 물론 특정 기능을 위해서는 해당 운영체제 전용으로 만들 필요도 있습니다만, 공통되고 평범한 기능이라면 표준을 이용하는 것이 바람직합니다.

ThingPlug에서는 oneM2M 표준을 지원하기 때문에 표준을 준수하는 사물과 서비스, 혹은 다른 IoT 플랫폼까지 별다른 수정 없이 연

동할 수 있습니다.

oneM2M은 TCP/IP 모델에서 애플리케이션 계층 Layer에 해당하는 표준으로 REST 아키텍처에 기반하여 만들어졌습니다.



표준에 포함된 내용은 IoT 서비스를 위한 다양한 리소스들을 정의하는 방법과 이를 다루는 방식 그리고 HTTP, MQTT, CoAP 같은 통신 프로토콜에 바인딩하는 방법 등입니다. ThingPlug를 이용하면 oneM2M 프로토콜에 기반한 IoT 서비스를 훨씬 쉽게 만들 수 있으며 원하는 통신 방식을 선택해서 사용할 수도 있습니다.

oneM2M에 대해서는 다음 장부터 상세한 설명이 이어지겠습니다만, 어떻게 IoT 서비스를 만들 수 있는지 실천해 보고 싶으시다면 개발자들의 소셜 네트워크 깃허브 [github](#)에 공개된 ThingPlug 스타터킷 **Starter Kit*1**을 통해 바로 체험해 볼 수도 있습니다.

» 안정성이 입증된 GMMP

두 번째 방식은 SK텔레콤의 독자적 프로토콜인 GMMP **Global M2M Protocol**와 REST API를 이용해서 IoT 서비스를 개발하는 것입니다.

oneM2M이 국제 표준이지만 최초 릴리즈 **Release 1**가 최근에만 공개되었고 현재 진행형으로 발전하는 표준이기 때문에 아직 oneM2M 상용 사례는 많지 않습니다. 따라서 oneM2M이 완전히 확산되기 전까지는 IoT 플랫폼들 저마다의 자체 프로토콜도 계속 활용될 것으로 보입니다.

GMMP는 IoT 이전 M2M 시절부터 디바이스와 플랫폼 사이의 통신을 위해서 애용된 프로토콜로 이미 많은 수의 다양한 상용 서비스에서 활발하게 사용하고 있는 성숙한 프로토콜입니다.

GMMP는 디바이스 등록, 주기 정보나 이벤트성 정보 전달, 제어 메시지 수신 등 IoT 서비스 개발을 위한 다양한 기능들을 이미 갖추고

*1 URL = <https://github.com/SKT-ThingPlug/thingplug-starter-kit>

있습니다. 게다가 여러 상용 서비스를 통해 편리성과 안전성이 입증된 상태입니다. 특히 TCP 기반의 프로토콜이기 때문에 HTTP 클라이언트가 올라가기 어려운 저성능 디바이스에서도 쉽게 구현해 탑재할 수 있다는 장점을 갖고 있습니다.

디바이스 쪽에서는 GMMP를 통해 플랫폼과 연결하여 데이터를 올리고 제어 명령을 받는다면 서비스 앱과 ThingPlug 사이의 인터페이스는 REST API가 담당하고 있습니다. REST API는 일반적으로 HTTP 프로토콜을 통해 데이터를 전달하거나 가져오는 API로, 웹 서비스에서 가장 많이 쓰이는 형태의 API입니다.

ThingPlug는 디바이스에서 GMMP로 올려준 정보를 REST API를 통해 서비스 앱에서 가져갈 수 있도록 제공하고 있으며 제어 명령 전달도 REST API 호출을 통해 가능합니다. 따라서 디바이스에서 GMMP로 데이터를 플랫폼에 전달하면 REST API를 통해 이 정보를 받아와 서비스 앱으로 보여주고, REST API로 제어 명령을 내리면 GMMP를 통해 디바이스를 제어하는 형태로 IoT 서비스 개발이 가능합니다.

GMMP의 특징 중 하나로 콘텐츠 내용에 제약을 두지 않는다는 점이 있습니다. 디바이스와 서비스앱은 어떠한 콘텐츠건 규격에 대한 제약 없이 ThingPlug를 통해 주고 받을 수 있습니다. 이러한 특성 때문에 GMMP는 다양한 서비스로 확장이 가능한 유연한 방식입니다. 그리

고 많은 굵직한 응용 사례들이 오랜 기간 안정적으로 운영 중인 안정성이 검증된 방식입니다.

만약 구현하려는 아이디어가 국제 표준만으로는 충분하지 못하다면 GMMP를 살펴보는 것이 좋습니다. TCP/IP로 연결하기 위해 필요한 라이브러리 등을 다양하게 지원하고 있으며, 서비스 앱은 REST API를 활용하면 IoT 서비스를 쉽게 만들 수 있습니다.

» 하루만에 똑딱, ThingPlug 서비스 플랫폼

세 번째 방식은 SK텔레콤 ThingPlug의 특별한 기능 중 하나인 ‘Thing Plug 서비스 플랫폼’을 사용하는 방식입니다.

IoT 플랫폼 중에서 서비스 플랫폼이란 서비스 앱에 다양한 기능들을 제공하는 플랫폼을 의미합니다. ThingPlug의 서비스 플랫폼은 대부분의 IoT 서비스들에서 공통적으로 사용하는 기능들을 미리 구현해 놓아, 프로그래밍을 하지 않고도 IoT 서비스를 개발할 수 있도록 지원하고 있습니다.

ThingPlug 서비스 플랫폼은 클라우드 환경에서 SaaS **Software as a Service** 형태로 서비스됩니다. 만들고자 하는 IoT 서비스가 매우 특화된 기능들이 필요한 경우가 아니라면 미리 만들어진 ThingPlug 서비스 플랫폼의 기능들을 활용해서 IoT 서비스를 개발하고 운영하는데 드는 비용을 크게 줄일 수 있습니다.

ThingPlug 서비스 플랫폼에서 제공하는 서비스 포털은 센서 데이터 조회, 디바이스 상태 모니터링 등의 기본적 기능부터 시계열 데이터 분석, IFTTT 형태의 룰 기반 이벤트 처리 등 고도화된 기능까지 클릭 몇 번으로 사용할 수 있는 웹 기반 서비스입니다. 일반적으로 대규모 서버 구축과 개발을 통해서만 사용할 수 있는 이러한 기능들을 미리 구현해서 쉽게 쓸 수 있도록 제공하고 있기 때문에 IoT 서비스를 만드는 사람들에게 실용적인 도움이 됩니다.

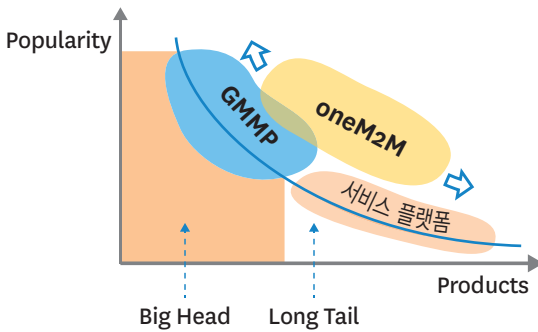
서비스 플랫폼을 사용하는 대표적인 방법은 오픈소스 하드웨어를 사용하는 것입니다. ThingPlug 서비스 플랫폼에서는 라즈베리 파이, 아두이노, 비글본, 엠베드 등 다양한 오픈소스 하드웨어를 지원하는 미리 만들어진 프로그램들을 제공하고 있어, 연동 가이드에 따라 설치만 하면 어떠한 추가 작업 없이도 바로 IoT 서비스를 만들 수 있습니다. 다양한 센서를 취향에 맞게 조립하고 연동한 뒤 클릭 몇 번을 통해 대시보드를 꾸미는 것만으로 IoT 서비스가 똑딱 만들어지는 것입니다.

전기 제품의 플러그를 콘센트에 끼우는 것이 누구나 할 수 있는 일인 것처럼, 오픈소스 하드웨어를 ThingPlug에 연결해서 IoT 기기로 만드는 것도 이제 이쯤처럼 쉽게 할 수 있습니다.

이상으로 ThingPlug를 사용하는 세 가지 방식에 대해 간단히 알아보았습니다. 각 방식에 대한 자세한 설명과 방법에 대해서는 뒤에서 본

격적으로 알아보도록 합시다.

그렇다면 ThingPlug를 사용하려고 하는 사용자는 어떠한 기준으로 세 가지 방식 중에 하나를 선택하면 될까요? 사용하기 편한 방법으로, 마음에 드는 프로토콜을 원하는 대로 선택해서 개발하셔도 상관없습니다. ThingPlug는 사용자가 원하는 방식을 마음대로 선택할 수 있도록 열려 있기 때문입니다. 하지만 특별히 선호하는 방식이 없다면 다음과 같은 기준으로 자신의 IoT 서비스에 맞는 방식을 선택하는 것을 추천합니다.



국제 표준에 기반한 IoT 서비스를 만들고 싶거나 MQTT, CoAP 기술을 활용하고 싶다면 oneM2M을 선택하는 것이 좋습니다. oneM2M은 지속적으로 성장하는 국제 표준이기 때문에 앞으로 oneM2M을 지원하는 플랫폼과 IoT 장치들은 점점 더 많이 나올 것입니다. 따라서 oneM2M을 이용해서 서비스를 만든다면 추후에 표준을 지원하

는 다른 플랫폼이나 단말들과 쉽게 연동할 수 있을 것입니다. 또한 oneM2M에서는 MQTT, CoAP 등 IoT에 적합한 통신 프로토콜을 지원하고 있습니다. 이런 IoT 특화 기술을 활용하고 싶다면 oneM2M을 통해 ThingPlug를 활용하는 것이 적합합니다.

oneM2M은 최초 릴리즈가 2015년 1월에 나온 최신의 프로토콜입니다. 따라서 신기술을 적극적으로 도입하고 빠르게 발전하고 있지만

BOX

MQTT(Message Queuing Telemetry Transport)

MQTT는 열악한 네트워크 환경에서 대량의 장치에 대한 모니터링을 목적으로 데이터 수집을 용이하게 하기 위해서 IBM이 1999년에 개발한 메시지 프로토콜입니다.

메시지 전송 패턴이 Request-Response 형태인 CoAP이나 HTTP와 달리 MQTT는 Publish-Subscribe(PUB-SUB) 패턴 개념 위에 개발된 프로토콜입니다. PUB-SUB 메시지 패턴은 다양한 일간지 대리점에서 사용하는 형태라고 할 수 있습니다. 매일 아침 신문 신간이 나오면, 신문사는 간단히 신간을 각 지역에 있는 대리점에 전달(Publish)하고, 해당 대리점은 사전에 구독 계약이 되어 있는 계약자들에게 신간을 전달(Subscribe)하는 형태입니다. 위에서 언급했던 바와 같이 PUB-SUB 전송 패턴에서는 대리점과 같은 역할을 수행하는 Message Broker가 존재합니다. MQTT도 PUB-SUB 패턴을 추구하는 프로토콜이기 때문에 Message Broker를 네트워크에 포함하여야 합니다.

Response 전송 패턴이 커플 사이에 발생하는 1:1 통신에 적합하다면, PUB-SUB은 위의 예에서 언급했던 바와 같이 1:N 또는 N:N에서 좀 더 적합한 전송 패턴이라고 볼 수 있습니다.

MQTT는 저전력, 전송 지연 및 손실이 발생하는 열악한 네트워크 환경을 고려하였습니다. 요구사항에 따라 상이한 QoS 레벨을 정의하고 있어, 메시지의 전달에 대한 신뢰성을 보장할 수 있는 방식도 제안하고 있습니다.

아직은 실제 대규모 서비스에서 사용하고 있는 상용 사례는 부족한 것이 사실입니다. 하지만 앞의 그림처럼 빅헤드 Big Head와 롱테일 Long Tail을 모두 커버할 수 있는 IoT 서비스를 위한 범용적인 프로토콜이기 때문에 대부분의 IoT 서비스에 적합하며 oneM2M을 통해 ThingPlug를 사용한다면 국제 표준을 사용하는 이점을 충분히 얻을 수 있을 것입니다.

IoT 플랫폼에서 디바이스 연결 관리나 안정적인 서비스만 제공해 주기를 원한다면 IoT 서비스에 맞는 디바이스와 서버를 입맛에 맞게 구성할 수 있는 GMMP를 추천합니다. GMMP는 SK텔레콤이 꾸준히

BOX

CoAP(Constrained Application Protocol)

CoAP은 기존의 HTTP를 IoT 시대에 요구되는 프로토콜 사항을 만족하도록 개선하기 위해서 개발된 프로토콜입니다. 특히 REST 지원과 메시지 경량화라는 요구사항을 충족시키기 위해 인터넷 표준화 단체인 IETF 내 코어 워킹그룹(Core WG)에서 만들었습니다.

RESTful 자원(Resource)에 접근이 용이하도록 CRUD(Create, Retrieve, Update and Delete) 동작을 기본적으로 지원합니다. 또한, 경량화 목표를 달성하기 위해 신뢰성(Reliability)을 사용자가 선택할 수 있도록 하고 있습니다. 하위 프로토콜로 TCP 대신 메시지 교환 신뢰성을 보장할 수 없는 UDP를 채택하였으며, 별도의 메시지 신뢰성을 보장받기 위하여 메시지 교환 타입을 확인형(CON)으로 선택할 수 있습니다. 불필요한 기능을 최대한 배제한 CoAP의 헤더의 크기는 약 10~20바이트로 기존 HTTP 대비 10%로 줄어듭니다. 이로 인하여 IoT 환경에서 빈번히 발생할 수 있는 작은 크기의 제어 메시지들을 보다 효율적으로 교환할 수 있게 됩니다.

발전시켜 온 프로토콜로 이미 많은 상용 사례가 있는 방식입니다. 따라서 이미 검증된 안정적인 서비스가 가능하고 높은 자유도를 가지고 서비스를 만들 수 있습니다. 통신사 서비스와의 연결을 기본 API로 제공하는 점도 GMMP의 강점입니다. 특히 유심^{USIM} 카드가 포함된 디바이스를 고려하고 있다면 GMMP를 사용하는 경우 많은 이점이 있습니다.

GMMP의 안정성과 자유도는 빅헤드에 해당하는 IoT 서비스에 적합한 경우가 많습니다. 대규모 사업의 경우 특화된 기능을 쓰는 경우가 많고 안정적인 서비스를 제공하는 것이 무엇보다 중요한데 다양한 기능의 API를 제공하고 안정적인 상용 사례를 많이 가지고 있는 GMMP는 이에 딱 맞는 방식입니다. 또한 TCP 기반의 프로토콜이기 때문에 저성능의 IoT 디바이스에서도 별 무리 없이 사용할 수 있어 대규모로 저성능의 디바이스를 배포하는 경우에도 쓸 수 있는 방식입니다.

오픈소스 하드웨어를 통한 빠른 서비스 구성이나 프로토타입을 만들고 싶은데 서비스를 위한 별도 서버를 구성하는 것이 부담된다면 ThingPlug 서비스 플랫폼을 사용하는 것이 좋습니다. 대부분의 IoT 서비스에 필요한 기능들을 이미 갖추어 놓았고 프로그래밍을 하지 않고도 누구나 쉽게 사용할 수 있도록 마련되어 있습니다. 따라서 DIY를 하고 싶어 하는 개인 사용자나 롱테일을 대상으로 하는 IoT 서비스에 적합한 방식입니다.

서비스 플랫폼에서 제공하고 있는 기능은 자유도가 조금 떨어질 뿐 매우 강력합니다. 빅헤드에 해당하는 IoT 서비스라도 디바이스 개발에 집중하고 서비스 서버는 이미 만들어져 있는 기능을 사용하고 싶다면 충분히 실제 서비스에 사용 가능합니다.

ThingPlug는 위에서 설명한 방식을 기본으로 다양한 형태로 좀 더 쉽게, 좀 더 빠르게 IoT 서비스를 만들 수 있도록 계속해서 진화 중인 플랫폼입니다. 따라서 IoT 서비스를 만들어 보려고 하시는 분들 모두에게 ThingPlug는 다른 얼굴로 비칠 것입니다. 자신에게 꼭 맞는 시나리오의 ThingPlug를 이용해서 IoT 서비스를 만든다면 분명 적잖은 비용과 시간을 아낄 수 있을 것입니다.

ThingPlug의 장점

그렇다면 ThingPlug가 다른 IoT 플랫폼들과 비교해서 차별화되는 부분은 어떤 것이 있을까요? IoT 플랫폼으로서 ThingPlug가 추구하고 있는 방향과 차별화된 장점들을 살펴보면 플랫폼을 선택하는데 도움이 될 것입니다. ThingPlug의 차별점 중 주목할만한 것으로는 다음을 들 수 있습니다.

» 개발 및 운영 비용 절감

ThingPlug는 범용 IoT 플랫폼으로 다양한 서비스를 수용할 수 있도록 데이터 저장구조가 유연하고, 인증, 데이터 수집, 상태관리, 제어 등의 공통적인 기능을 오픈 API를 통해 쉽게 이용할 수 있도록 제공하고 있습니다. 또한 ThingPlug 서비스 플랫폼을 사용하면 추가적인 서비스 앱 개발 없이 IoT 서비스를 구성할 수도 있습니다.

일반적으로 서비스를 개발할 때 구현해야 하는 부분을 데이터, 화면, 비즈니스 로직으로 나누어 볼 때, 세 영역 모두에서 ThingPlug를 통한 개발 비용 절감이 가능합니다. 실제로 처음에는 6개월의 개발 기간이 필요할 것으로 예상했던 업체가 ThingPlug의 GMMP를 활용해 3개월 만에 목표한 시스템을 조기 개발 완료하고 상용화한 사례가 있습니다. 50%의 개발기간을 절감한 것입니다. ThingPlug에서 제공

CHAPTER

2

oneM2M 표준을 탑재한 ThingPlug



oneM2M 기반 플랫폼 ThingPlug
ThingPlug의 oneM2M 인터페이스 활용하기
IoT의 뜨거운 감자 '보안'



oneM2M은 IoT 플랫폼을 위한 국제 표준입니다. ThingPlug는 통신사 세계 최초로 oneM2M을 적용한 상용 플랫폼입니다. 향후 확산이 유력한 국제 표준을 적용함으로써 ThingPlug를 사용하는 IoT 서비스들이 표준을 준수하는 다른 사물이나 서비스와 손쉽게 연동할 수 있게 됩니다.

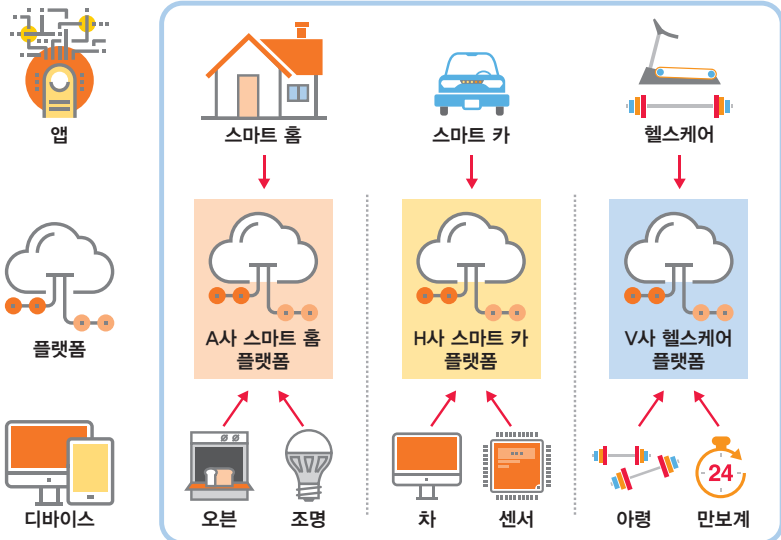
이번 장에서는 oneM2M 표준에 대해 자세히 알아보고 ThingPlug에서는 어떻게 oneM2M을 구현했는지 살펴보겠습니다.

oneM2M 기반 플랫폼 ThingPlug

IoT 서비스를 위해 등장한 플랫폼 표준, oneM2M

IoT로 할 수 있는 일은 무궁무진합니다. 하지만 지금까지 이 다양한 응용 분야를 지원하기 위한 IoT 플랫폼들은 기업에서는 특정 애플리케이션과 특정 장치를 위한 수직적 플랫폼(Vertical Platform)으로 설계되고 시장에서는 제한적으로 활용될 수밖에 없었습니다.

예를 들어 다음 그림과 같이 A사의 플랫폼은 ‘스마트 홈’을 위해서만 설계된 후 ‘A사의 특정 장치-오븐, 조명’의 연결만을 지원하였습니다.

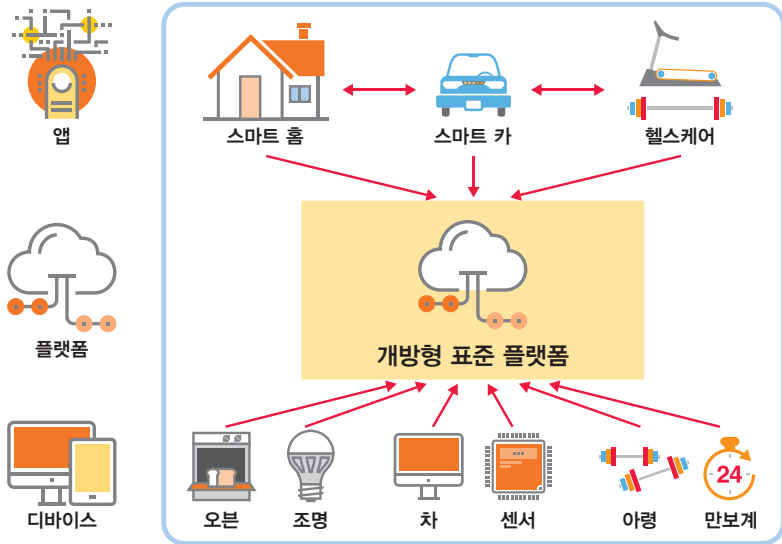


이 A사의 플랫폼으로는 스마트 카에서 필요한 요구사항을 만족할 수도 없고 스마트 카를 위한 다양한 장치와의 연결도 불가능합니다.

스마트 카의 경우 플랫폼이 먼저 챙겨야 하는 핵심은 안전과 관련된 사항일 것입니다. 스마트 카에서 핵심적인 요구사항 중 하나는 네트워크에서의 낮은 지연^{Low Latency}, 즉 빨리 반응하는 것입니다. 이를 보장하는 통신 기술이 필수적인데 해당 기술의 요구사항은 스마트 홈 시나리오에서는 큰 필요가 없었기에 아마 준비되어 있지 않을 것입니다. 이 경우 스마트 홈을 위해 설계된 플랫폼을 스마트 카에서 활용하기 위해서는 해당 플랫폼이 재설계되고 추가 기능을 탑재해야 합니다.

하지만 모바일 시대가 펼쳐진 후 IoT 시장에서도 다양한 기술적 요구사항을 당장 해소해 달라는 요구가 늘 수밖에 없었습니다. 이를 만족하기 위해서 다양한 응용 분야 간 IoT 융합이 가능한 수평적 플랫폼 **Horizontal Platform**들이 개발되고 있습니다. 다음 그림과 같이 수직적 플랫폼과 달리 수평적 플랫폼은 단일 플랫폼에서 다양한 응용사례 간의 융합이 가능하고 개방적으로 다양한 장치들이 수용될 수 있습니다.

하지만 완벽하게 수평적인 플랫폼을 개발한다는 것은 쉬운 일이 아닙니다. 쉽게 완성될 수 있는 것이 아니기에 여기저기서 많은 시도가 일어납니다. 수평적 플랫폼 기술의 예로는 무엇이 있을까요? IoT 영역에서 많이 이야기되고 있는 것으로는 퀄컴 주도하에 LG전자, 마이크로소프트 등의 업체가 연합하고 있는 올신 얼라이언스 **AllSeen Alliance**의



울조인 AllJoyn이 있습니다. 동시에 이를 견제하기 위해 삼성전자, 인텔의 주도하에 다양한 업체가 연합하고 있는 OIC(Open Interconnect Consortium) 또한 대표적 사례입니다.

언급된 기술들의 특징은 업체 주도의 기술이며, 업체들이 각자의 필요와 이해관계에 따라 연합했다는 것입니다. 이런 업체 주도 기술은 시장에 빨리 보급되고 기술이나 보안의 안정성이 확보될 수 있다는 장점이 있지만 다른 면에서는 기술 간 장벽을 세워 개방성은 부족할 수 있습니다.

이렇게 업체 간 기술 주도로 인해 시장 내에서 발생할 수 있는 파편화와 지역적으로 발생하는 IoT 플랫폼 기술의 불균형 문제를 해결하고

자 조직된 최초의 국제 표준 단체가 있습니다. 바로 oneM2M입니다.

2012년 7월에 설립된 oneM2M 표준의 핵심은 ICT 산업을 이끄는 대륙 또는 국가 내에서 독자적으로 개발하는 IoT 플랫폼을 통합하자는 것입니다. 그리하여 한국, 유럽, 북미, 중국, 일본, 인도를 대표하는 정부 주도의 표준화 단체 SDO, Standard Development Organization는 ‘각 대륙 또는 국가의 정부 주도의 표준화 단체 내에서 중복 기술 개발 금지 및 oneM2M 기술을 각 표준화 단체 내 권고 Recommendation IoT 플랫폼 기술로 비준 Ratification하는 것’에 합의했습니다. 그리고 oneM2M은 드디어 2015년 1월 oneM2M 최초 릴리즈를 비준하게 됩니다.

NOTE

권고(Recommendation) 기술의 의미는 말 그대로 권고하는 것이지만 합의한 대륙 또는 국가 내 기업이나 정부단체가 해당 기술을 따를 의무가 있는 것은 아닙니다.

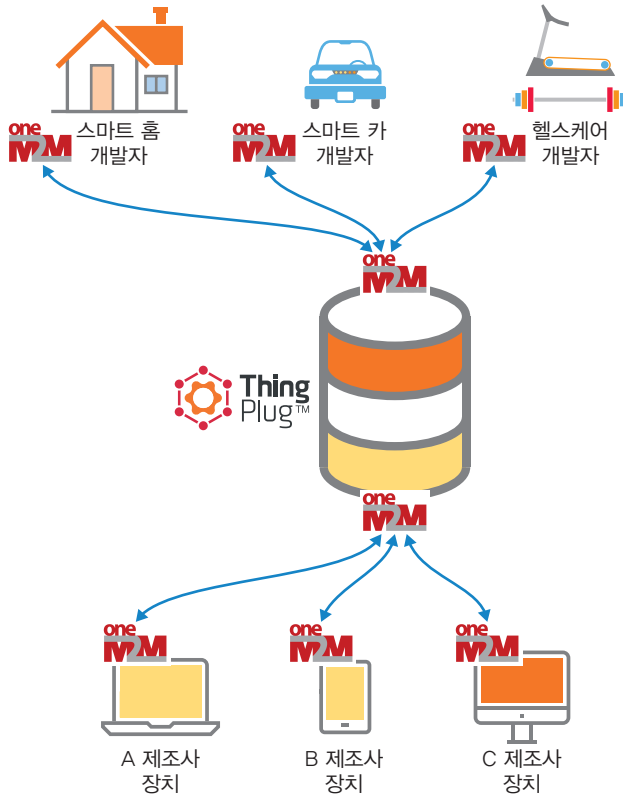
oneM2M 표준을 수용한 ThingPlug

oneM2M 표준은 ThingPlug에 다양한 제조사들의 장치들과 다양한 애플리케이션 개발자들이 접근할 수 있도록 하는 중요한 인터페이스 중 하나이며, ThingPlug 플랫폼을 풍요롭게 해주는 중요한 역할을 합니다. ThingPlug가 제공하는 oneM2M 표준 기술을 왜 주

목해야 할까요?

현재 사용되고 있는 다양한 IoT 플랫폼이 제공하는 가치 중 중요한 부분은 API를 통해 IoT 애플리케이션을 개발할 때 필요한 장치 등록, 데이터 공유, 장치 제어, 보안 등과 같은 기능을 제공한다는 것입니다. 개발자들은 플랫폼에서 제공하는 API를 통해 좀 더 쉽고 빠르게 IoT 애플리케이션을 개발할 수 있어 생산성이 향상됩니다. 그러나 만약 개발자들이 서로 다른 플랫폼에서 동작하는 애플리케이션을 개발해야 한다면 어떨까요? 플랫폼마다 서로 다른 API 활용방법을 모두 숙지하고 있어야 하므로 어찌 보면 더 큰 불편함을 느낄 수 있습니다. 그런데 이 API가 국제 표준과 산업 표준에 의해서 보편적으로 활용될 수 있다면 개발자들의 고충을 조금이나마 덜어줄 수 있지 않을까요?

IoT 장치 제조사는 IoT 장치가 다양한 플랫폼 사업자들에게 바로 연결되는 상호호환성 **Interoperability**을 유지할 수 있게 됩니다. IoT 개발자라면 한 번 배운 API를 더 다양한 응용 분야에 활용할 수 있게 됩니다. 또한, 이 IoT의 가치가 어떤 특정 시장이나 지역에서 멈추지 않고 글로벌하게 퍼지기 위한 발판이 될 것입니다.

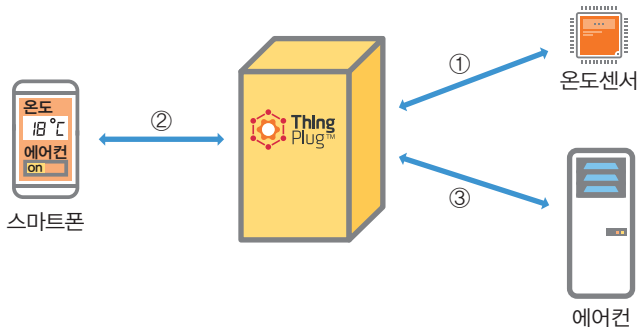


이처럼 국제 표준을 구현한 본격적 플랫폼의 등장이 IoT 생태계에 해 줄 수 있는 일은 적지 않습니다. oneM2M이 기대될 수밖에 없는 이유입니다.

ThingPlug에 탑재된 oneM2M

» ThingPlug에서의 oneM2M

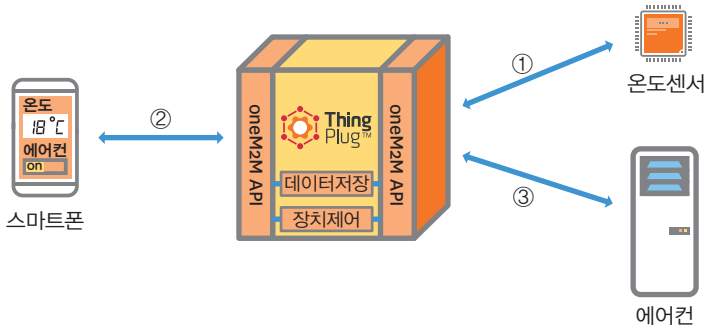
oneM2M 표준은 ThingPlug 플랫폼에서 어떤 역할을 하게 될까요?



위의 그림은 집에 있는 온도센서의 정보를 통해서 에어컨을 원격에서 제어할 수 있는 간단한 IoT 사례를 나타내고 있습니다. 이 사례의 전체적인 흐름은 다음과 같습니다.

1. 온도 센서는 5분을 주기로 온도를 측정하여 해당 온도 정보를 ThingPlug에 전달합니다.
2. 스마트폰에 설치된 IoT 애플리케이션은 ThingPlug로부터 온도센서가 측정한 온도 정보를 조회하여 화면에 보여줍니다.
3. 사용자의 조작 또는 미리 설정된 조건 정보에 따라 IoT 애플리케이션은 ThingPlug에 에어컨의 제어(예: 30도 이상 시, 에어컨 동작)를 요청합니다.

이를 위해 ThingPlug에서는 다음 그림과 같이 ‘데이터 저장’과 ‘장치 제어’ 기능을 위한 소프트웨어 기능-Function이 탑재되어 있고, 해당 기능을 스마트폰에 설치된 IoT 애플리케이션과 IoT 장치들이 접근할 수 있도록 API를 제공합니다. 중요한 부분은 ThingPlug의 내부 기능이 국제적으로 표준화된 API로 외부에 노출된다는 것입니다. 다른 측면으로 보자면, oneM2M API를 기반으로 개발된 애플리케이션이나 장치들이 그대로 ThingPlug의 다양한 기능을 사용할 수 있다는 뜻이 됩니다. 동시에 ThingPlug에서 개발된 애플리케이션이나 장치는 별도의 소스 코드 수정 없이 oneM2M API를 노출하는 다른 플랫폼에도 자연스럽게 연결될 수 있게 됩니다.

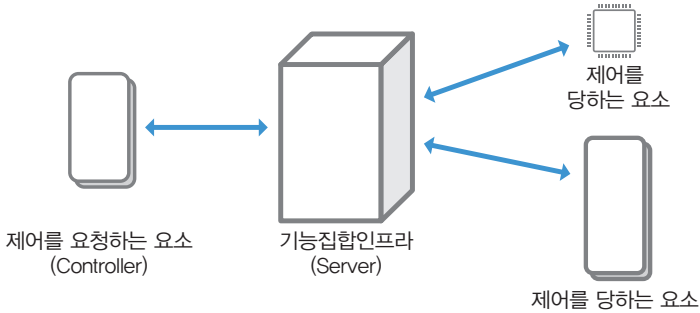


ThingPlug에 oneM2M 표준을 탑재하였다는 것은 oneM2M에서 정의한 기능을 노출할 수 있는 표준화된 API를 제공하며 또한 원격에 있는 API를 호출하기 위해서 규격화된 프로토콜을 제공한다는 것입니다.

그럼, ThingPlug에서 제공하는 oneM2M 표준 기능은 무엇이고 기능을 호출하기 위해 어떤 스타일의 API를 제공하는지, 그리고 원격 호출을 위한 통신 프로토콜은 어떤 프로토콜을 사용하는지 알아볼 순서입니다. ThingPlug에 적용된 oneM2M 표준의 주요 기능을 살펴보는 일은 ThingPlug의 전체 모습을 파악하는 데 꽤 중요한 일이기도 합니다.

oneM2M의 주요 구성 요소*1

ThingPlug에서 제공하는 oneM2M의 주요 기능을 살펴보기에 앞서 잠시 전체를 조망할 수 있도록 시스템 구성요소를 살펴보도록 하겠습니다. oneM2M 표준에서 이야기하는 기본적 IoT 시스템 구조는 제어를 하는 요소(예: 제어 앱)와 제어 요청에 따라 제어되는 요소(예: IoT 장치), 그리고 각 요소의 기능 요청을 수용하고 각 요소 간 연결을 유지해주는 서버로 구성됩니다.

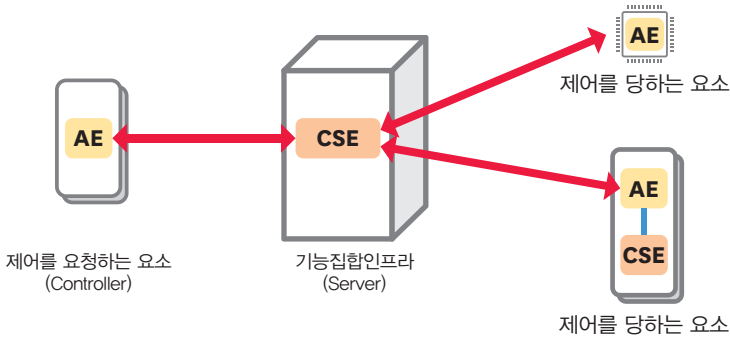


*1 관련표준: oneM2M TS-0001 (Functional Architecture): 5. Architecture Model

플랫폼 표준으로서 oneM2M의 핵심은 IoT 애플리케이션 개발자들이 중복 개발해야 하는 기능을 최소화하기 위해서 공통 기능을 제공하는 데 있습니다.

oneM2M 표준에서는 이 공통 기능 서비스를 제공해주는 실체를 CSE **Common Service Entity**라 칭하며, CSE는 공통 기능들을 API로 노출합니다. 또한, 해당 제공 기능을 활용하는 애플리케이션을 AE **Application Entity**라 부릅니다.

AE는 바로 우리가 만들고 싶은 IoT 앱입니다. AE는 CSE에서 제공하는 API를 통해서 공통기능을 활용하게 됩니다. oneM2M 표준은 공통 기능의 CSE와 CSE에 접근하는 인터페이스를 다루고 있습니다. 다음 그림은 oneM2M 세부 기능 영역을 포함한 IoT 시스템의 예를 도식화한 것입니다. 여기서 ThingPlug는 가운데 위치하는 서버라고 생각하면 됩니다. oneM2M의 공통 기능이 탑재된 CSE를 포함하고 있으며, 기능을 노출하는 다양한 기능의 API로 구성되어 있습니다.

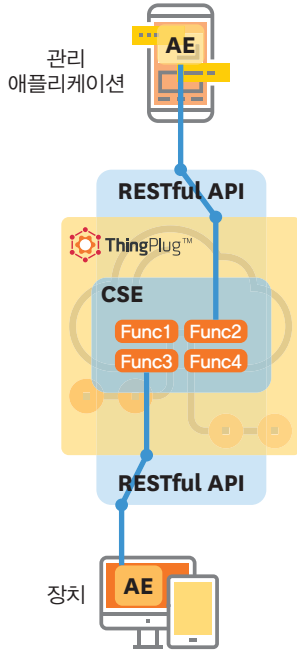


oneM2M CSE가 제공하는 공통 기능

그렇다면, oneM2M 표준의 핵심이 되는 CSE는 어떤 공통 기능을 제공할까요? 다음 표는 CSE가 제공하는 공통 기능입니다. 14가지의 공통 기능이 업계의 요구사항을 충족하기 위해 표준화되어 있습니다.

CSE가 제공하는 공통 기능	
데이터 관리	데이터 저장 및 관리, 데이터 분석 기능
연결제어	메시지 전달 관리 및 정책에 기반한 전송 QoS제어
구독/통지	정보 변경에 대한 구독/통지 기능
장치/서비스 등록	애플리케이션 및 장치 등록
그룹 관리	그룹 설정 및 다수의 장치 일괄 관리/제어
보안	단대단 보안 연결 제공, 인증/권한 설정 기능
정보 탐색	특정 정보 탐색/ 특정 정보에 대한 통지
이동통신망 연동	엑세스 네트워크(3GPP) 연동 기술
위치 관리	장치에 위치 정보 제공 및 관리
장치 관리	OMA DM, BBF TR-o6g 연동을 통한 장치관리 기능 제공
과금	서비스 계층 과금
시멘틱	자원 의미(시멘틱) 분석 및 검색 기술
이종 플랫폼 연동	oneM2M과 다른 플랫폼 간의 연동 기술
사업자 간 로밍	oneM2M 서비스 사업자 간의 로밍 기술

ThingPlug에 구현된 REST API의 기본 구조와 호출하는 방법*2



위의 그림에서처럼 CSE의 주요 기능은 REST API로 드러나고 있습니다. 제어를 하는 앱과 이 제어를 당하는 IoT 장치는 모두 REST API로 물리게 됩니다.

CSE에는 IoT에서 필요한 기능들이 구현되어 있고 각 기능은 한 개 또는 여러 개의 REST 기반 자원 API 형태로 연결되어서 기능을 호출할 수 있도록 구성되어 있습니다.

*2 관련표준: oneM2M TS-0001 (Functional Architecture): 8. Description and Flows of Reference Points)

원격에 위치한 IoT 장치가 ThingPlug에서 제공하는 oneM2M 기능에 접근하는 일은 웹의 상식을 따르고 있습니다.

분산형 시스템의 구조를 연구하는 엔지니어의 핵심 목표는 원격 장치에 정의된 기능을 쉽게 호출하는 것입니다. TCP/IP 등 복잡한 하위 계층에 대한 이해가 없더라도 개발자들이 분산형 시스템을 쉽게 구축할 수 있도록 해 주는 것이 목표라고 할 수 있습니다.

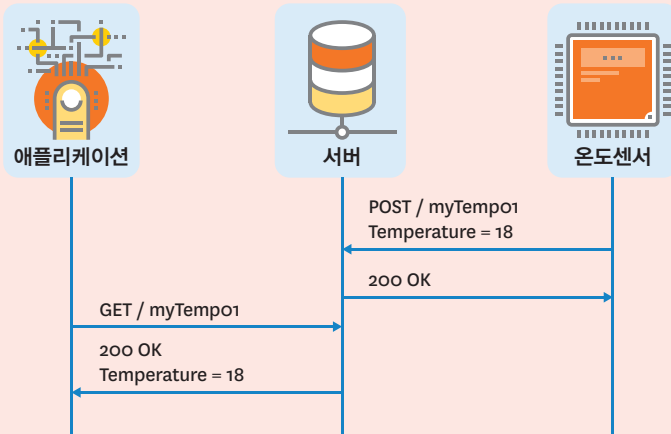
이를 위해서 지금까지 소위 RPC(Remote Procedure Call)라고 불리는 많은 원격 기능을 호출하는 API 구조들이 발표되었습니다. Java RMI, CORBA, DCOM, XML-RPC 및 SoAP이 이런 연구의 산출물입니다. 그 덕에 분산형 시스템들의 개발이 쉬워졌으며, 이는 M2M 시대에 장치의 기능을 호출하는 데 사용되곤 했습니다.

하지만 IoT 시대가 도래하면서 현재까지 가장 성공한 분산형 시스템인 웹 기술로 물리적 장치의 상태 제어를 연동하는 분산형 시스템으로 귀결되는 형태를 보이게 되었습니다. 이 API 구조는 이미 2000년 로이 필딩(HTTP의 저자 중 한 사람)의 논문으로 발표되었습니다. REST(Representational State Transfer)라는 API 구조를 처음 천명한 역사적 논문이었습니다. 물리적 장치의 기능들을 상태(State)가 정의된 자원(Resource)으로 연결하고, 기본적으로 제공되는 호출 메시지에 해당 자원의 상태를 조작하는 명령을 CRUD(CREATE, RETRIEVE, UPDATE, DELETE)라는 비교적 단순한 형태로 정의하였습니다.

REST의 API 철학은 웹에서 사용되는 프로토콜인 HTTP와도 잘 어울립니다. 예를 들어 웹에서는 물리적 또는 논리적 개체들이 식별자(URI)를 가지는 자원(Resource)으로 나타나고, 각 자원에 접근할 때는 CRUD로 매칭되는 POST, GET, PUT, DELETE 명령을 사용하기 때문입니다.

다음 그림과 같이 IoT 장치는 센싱 정보를 저장하는 기능을 웹 URI(/data-Storage)로 호출하고, 호출 시 해당 호출에 대한 동작을 데이터를 저장하기 위한 POST로 나타냅니다. 이와 반대로 저장된 센싱 정보를 조회하기 위해서는 동일한 웹 URI에 GET으로 호출하여 저장된 데이터를 쉽게 조회합니다. 또한, 웹 환경에 친숙한 REST API는 HTML5 표준을 통해서 다양한 애플리케이션과 연

계되어 좀 더 풍성한 IoT 앱을 개발하는데도 쉽게 확장될 수 있습니다.



REST의 기법은 간단하다는 장점이 있기 때문에 IoT 애플리케이션뿐만 아니라, 개방형 웹 API에서도 채택되어 사용되고 있습니다. 기존의 다른 API 기법인 SoAP, XML-RPC 등이 사용하기 복잡한 데 비해 REST 기법은 매우 간단하여 특히 개방형 웹 API에 잘 어울립니다.

oneM2M 표준화 논의 시에도 이런 REST의 장점을 인식하고 oneM2M 기능을 원격에서 호출하는 API 스타일을 REST로 진행하기로 합의하였습니다. oneM2M의 모든 장치 및 데이터에 접근하는 API는 자원의 형태로 식별되고, 해당 자원은 상태로 표현되며 해당 상태는 CRUD라는 명령으로 조작됩니다.

우선 호출 메시지의 패턴에 대해서 알아보시다. oneM2M 표준에 정의된 API를 호출하기 위해서는 약속된 형태의 통신을 해야 합니다. 이때 통신 방식은 패턴이 있는데 요청 **Request**과 응답 **Response**이 쌍을 이룹니다. 즉, 발신자 **Originator**가 해당 기능을 제공하는 수신자 **Receiver**에 요

CHAPTER

5

코딩할 줄 몰라도 가능한 IoT?



ThingPlug 서비스 플랫폼
ThingPlug 서비스 플랫폼의 특징 및 기능
ThingPlug 서비스 플랫폼 연동방법
ThingPlug 서비스 플랫폼으로 당장 이렇게 만들 수 있다



ThingPlug 서비스 플랫폼을 사용한 IoT 서비스 개발은 앞서 살펴본 oneM2M이나 GMMP를 사용한 IoT 서비스 개발과는 좀 다릅니다. 서비스 플랫폼에서는 다양한 IoT 서비스들에서 공통으로 많이 쓰이는 기능들을 미리 구현해 클라우드 서비스로 제공하고 있습니다. 또한, 오픈소스 하드웨어(Opensource Hardware)를 바로 연동할 수 있는 소프트웨어 패키지도 제공합니다. 서비스 플랫폼을 잘 활용하면 코딩 한 줄 하지 않고도 누구나 자신만의 IoT 서비스를 만들어 볼 수 있습니다.

IoT 서비스를 만들고 싶은데 서비스 운영에 대한 부담을 가지고 있는 분이나 오픈소스 하드웨어를 이용한 DIY(Do It Yourself) 개발에 관심이 있는 개인 사용자라면 ThingPlug 서비스 플랫폼이 많은 것을 도와줄 수 있습니다.

ThingPlug 서비스 플랫폼

IoT 이전 M2M 시절, M2M 플랫폼의 가장 중요한 기능은 연결 **Connectivity** 을 제공하는 것이었습니다. 각종 M2M 기술들로 사물들이 연결되면서 기존에는 할 수 없었던 원격 모니터링이나 제어가 가능해졌습니다. 이때의 플랫폼은 사물과 사물의 연결이 끊기지 않고 정보를 잘 전달할 수 있도록 연결을 관리하는 기능이 가장 중요했습니다. 특히나 통신사의 플랫폼들은 무선 통신망에 연결된 M2M 장치들의 연결이 끊기지 않도록 관리하는 다양한 기능을 제공하고 있었습니다.

물론 IoT에서도 연결은 가장 중요한 기능 중 하나입니다. 하지만 IoT에서는 M2M에서처럼 사물과의 통신을 통한 단순한 정보 교환에 머물지 않고 그로 인해서 생겨날 수 있는 다양한 '서비스'로 무게 중심이 이동하고 있습니다. 장치에서 정보를 수집하고 제어 명령을 전달하는 것 자체가 목적이었던 시대에서 연결된 장치와 수집된 정보들을 바탕으로 새로운 가치를 창출하고 서비스를 만드는 것이 중요한 시대로 바뀌고 있습니다. 시대 변화에 맞춰 기존의 M2M 플랫폼들도 IoT 플랫폼으로 진화하면서 IoT 서비스를 지원하기 위한 다양한 기능을 갖춰나가고 있습니다. API나 SDK를 통해 서비스 앱을 개발할 수 있게 지원하거나 아예 클라우드 기반의 IDE를 제공하여 개발을 지원하는 것이 그 예입니다.

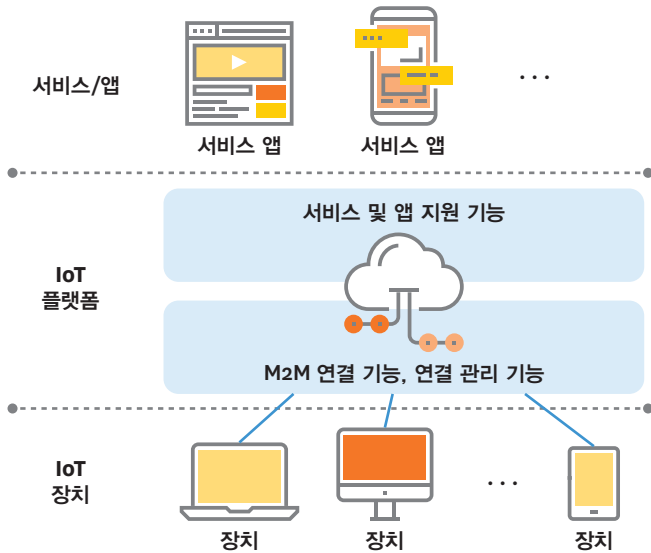
ThingPlug도 다양한 라이브러리나 SDK, API를 제공해서 IoT 서비스를 쉽게 만들 수 있도록 지원하고 있습니다.

하지만 대부분의 IoT 플랫폼에서 지원하는 기능들은 모두 개발자를 위한 것이기에 기술을 알아야 사용할 수 있습니다. 잘 만들어진 API, 사용하기 편한 라이브러리를 제공하더라도 누군가는 그것을 활용해 추가 개발을 해야만 IoT 서비스를 만들 수 있습니다.

ThingPlug의 ‘서비스 플랫폼’은 다릅니다. IoT 서비스에 적합한 사물만 개발하면 나머지는 ThingPlug에서 클라우드 서비스로 제공합니다. 그리고 이 기능은 중소규모의 IoT 서비스라면 추가적인 개발이 필요 없을 정도로 충분합니다. 오픈소스 하드웨어를 이용해 사물을 만든다면 개발자가 아니라도 간편하게 자신만의 서비스를 만들어 볼 수도 있습니다. 코딩을 할 줄 몰라도 IoT 서비스를 구성할 수 있는 것입니다.

서비스란 규모도 그 영역도 다양할 수밖에 없습니다. IoT 역시 예외가 아닙니다. 따라서 IoT 플랫폼은 플랫폼을 사용하는 IoT 서비스가 특별한 제약 없이 원하는 대로 서비스를 구성할 수 있도록 최대한의 자유도를 제공하기 위해 노력합니다.

하지만 어떤 경우에는 서비스를 개발하는 입장에서 이런 자유도가 부담될 수도 있습니다. 원하는 대로 개발할 수 있다는 의미는 디바이스부터 서비스 앱까지 전부 개발하고 운영까지 해야 한다는 뜻일 수



도 있기 때문입니다. 특히 아이디어를 빠르게 시제품으로 만들어 보려 하는 개인 개발자나 서버 운영에 대한 부담을 덜고 싶어 하는 작은 기업들이라면 오히려 어느 정도 제약이 있더라도 개발 및 운영에 대한 공수를 줄일 수 있는 지름길을 더 선호할 수 있습니다. 이 지름길에 해당하는 것이 바로 ThingPlug의 서비스 플랫폼입니다.

ThingPlug 서비스 플랫폼은 IoT 영역에서 공통으로 사용되는 서비스 기능들을 미리 구현해서 바로 사용할 수 있게 제공하고 있습니다. 미리 만들어져 있는 기성품이라서 맞춤의 느낌은 다소 떨어질 수 있지만, 대신 매우 빠르게 서비스를 만들 수 있습니다. 그간 다양한 영역의 서비스를 지원한 경험에 입각하여 IoT 서비스에 필요한 다양한

ThingPlug^로 시작하는 IoT 서비스개발

초판 1쇄 발행일 2015년 11월 27일

지은이 고가람, 안홍범, 김규백, 이종은, 이상민, 이재한
펴낸이 박지하
펴낸곳 페이지블루
등록번호 2014-000157
팩스 0504 056 9006
전화 070 8878 2325
전자우편 welcome@pageblue.kr

ISBN 979-11-954164-4-8 95000

(C) 고가람, 안홍범, 김규백, 이종은, 이상민, 이재한, 2015

Published by Pageblue, Printed in Korea

* 저작권법에 따라 보호받는 저작물이므로 무단 복제와 전재를 금합니다.

* 책값은 뒷표지에 있습니다.

* 잘못 만들어진 책은 구입처에서 교환해 드립니다.

ThingPlug로 시작하는 IoT 서비스개발

IoT 서비스개발도 플러그를 끼우듯 간단하게!

이 책은 사물인터넷 국제 표준인 oneM2M을 세계 최초로 상용화한 'ThingPlug 플랫폼'에 대한 가이드 북으로, 사물인터넷에 대하여 소개하며 누구든 손쉽게 IoT 애플리케이션을 개발할 수 있도록 안내해준다. 사물인터넷에 관심을 가지고, 직접 무언가를 해보고 싶은 모두를 위한 필독서!

- 송재승, 세종대학교 교수, oneM2M Testing WG 의장

이 책의 걸모습은 IoT 서비스 개발자를 위한 ThingPlug 플랫폼 가이드이지만, 행간에는 SK텔레콤이 긴 시간 동안 고민해 온 '미래를 향한 새로운 사회기술 시스템'으로서의 인터넷과 IoT에 대한 통찰이 가득하다.

- 장중혁, ATLAS Research&Consulting 부사장

ThingPlug는 생활 가치를 혁신할 수 있는 IoT 서비스들이 많이 만들어지기를 바라며 SK텔레콤의 경험과 기술을 집대성한 IoT 플랫폼입니다. 이 책이 많은 개발자 분들이 창조경제를 이끌어가는 IoT서비스를 쉽고 빠르게 구현할 수 있도록 지원하는 지침서가 되기를 기대합니다.

- 최진성, SK텔레콤 종합기술원 원장



정가 15,000원

ISBN 979-11-954164-4-8