



Interworking Proxy Entity(IPE): Kerlink(LoRa) GW

Software Version: 1.0.0

Develop Guide v1.0

Document Release Date: February 1, 2018

Software Release Date: February 1, 2018



Copyright (c) 2014, OCEAN
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Contents

1. Introduction.....	4
1.1. oneM2M Kerlink(LoRa) IPE.....	4
1.2. oneM2M Kerlink(LoRa) IPE Environment.....	4
2. IPE Sample Hardware.....	5
2.1. Introduce.....	5
2.2. Hardware Connectivity.....	5
3. IPE Sample Software.....	6
3.1. Download.....	6
3.2. Import project.....	8
3.3. Execution.....	9
4. Test.....	11
4.1. Kerlink(LoRa) IPE test.....	11

1. Introduction

1.1. oneM2M Kerlink(LoRa) IPE

oneM2M 표준 서버 플랫폼과 LoRa 기반 디바이스를 연동하기 위한 IPE(Interworking Proxy Entity)로서 Kerlink(LoRa) 게이트웨이 지원을 위한 기능을 탑재하여 게이트웨이 하부에 연결된 LoRa 디바이스를 oneM2M 서버 플랫폼을 통해 양방향(데이터수집/제어) 통신 지원을 목적으로 개발되었음.

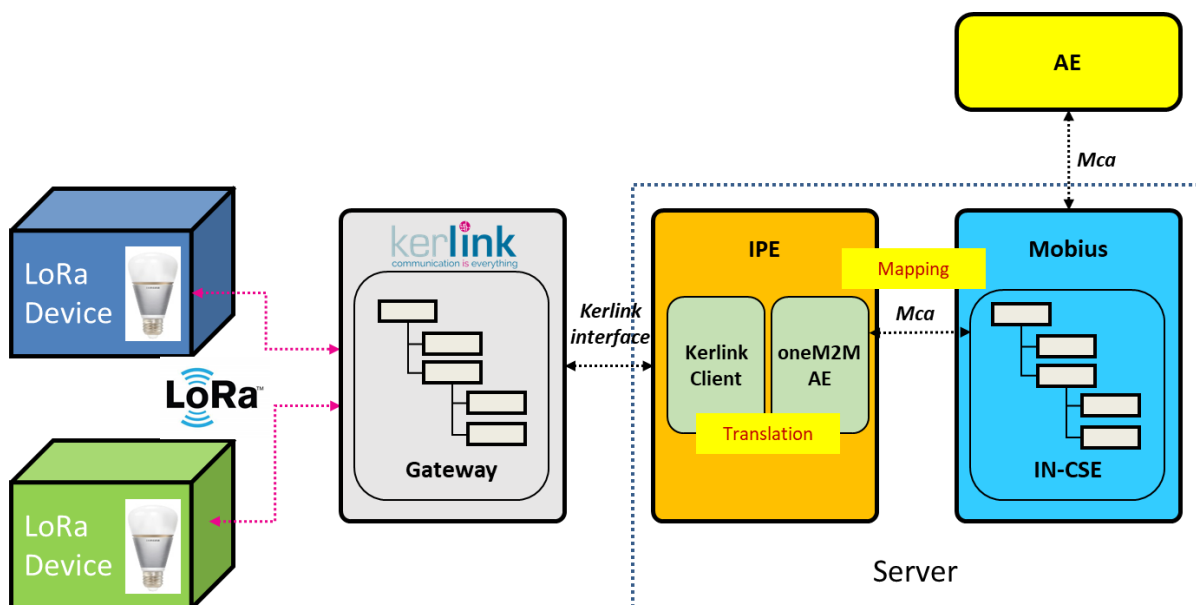


Figure 1 oneM2M Kerlink(LoRa) Interworking

1.2. oneM2M Kerlink(LoRa) IPE Environment

Kerlink(LoRa) IPE는 Java 언어를 활용하여 Eclipse IDE를 기반으로 개발되었으며 구동에 필요한 라이브러리를 포함하여 JVM(Java Virtual Machine)이 설치된 Windows 기반의 PC, Unix 기반의 PC 혹은 Raspberry Pi 등 임베디드 장치에서 구동이 가능함.

Kerlink(LoRa) 인터페이스를 활용한 파트의 경우 별도 제공 라이브러리가 아닌 Apache HTTP 라이브러리를 활용한 직접 개발을 통하여 구동 호환성을 제공토록 함.

2. IPE Sample Hardware

2.1. Introduce

Kerlink(LoRa) IPE는 기본적으로 LoRa 디바이스를 연동하기 위한 게이트웨이와 통신하는 것으로 그림 1과 같이 LoRa 디바이스 - 게이트웨이 제품이 필요함.

본 가이드에서는 Kerlink사의 IoT Station (SPN version)을 활용하였으며 LoRa 단말의 경우 (주)솔루엠의 LoRa Module을 활용하여 구성하는 것을 기준으로 작성되었음.



Figure 2 Kerlink IoT Station, 솔루엠 LoRa Module

2.2. Hardware Connectivity

커링크(Kerlink) 게이트웨이는 하부에 LoRa 표준을 지원하여 디바이스 연동 기능을 제공하며 SPN(Small Private Network) 서버 기능을 통해 웹을 통해 접속하여 연결된 디바이스의 상태 및 디바이스로부터 수신된 데이터 확인, 디바이스 제어가 가능함

그림 3과 같이 Kerlink 게이트웨이의 Lan 인터페이스에 케이블 연결 후 PC를 통해 게이트웨이의 SPN에 접속함.



Figure 3 Kerlink 게이트웨이 셋업

SPN이 제공하는 LoRa 모트 등록기능을 활용하여 LoRa 디바이스에 대한 등록을 수행함. 등록된 LoRa 모트의 정보 및 데이터는 그림 4와 같이 SPN에서 확인이 가능함.

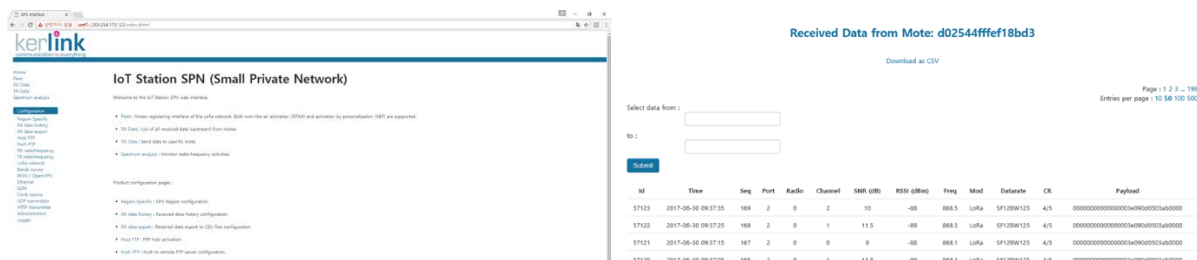


Figure 4 Kerlink 게이트웨이 SPN 접속 및 LoRa 모트 정보 확인 페이지

상세한 게이트웨이 활용 정보는 Kerlink에서 제공하는 IoT Station 매뉴얼 참조.

3. IPE Sample Software

3.1. Download

1. Download the Kerlink(LoRa) IPE Sample source from [IoT Ocean home page](#).

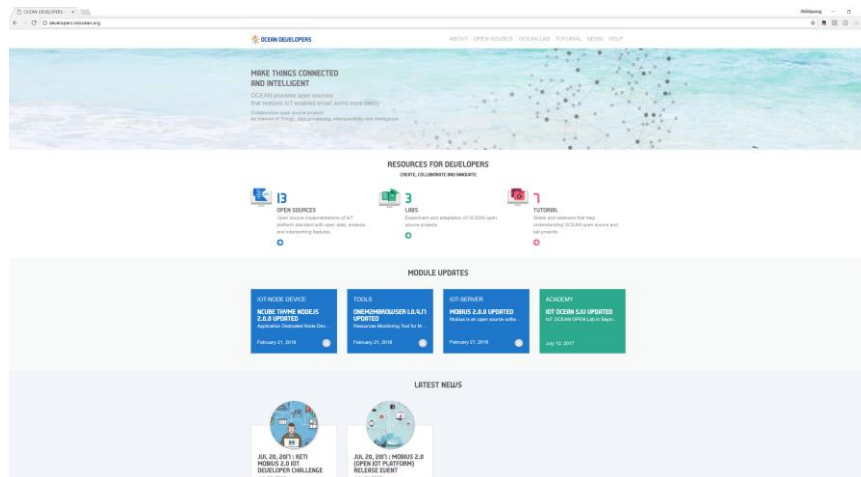


Figure 5 IoT Ocean home page

2. Download the Eclipse Neon from [Eclipse home page](#).

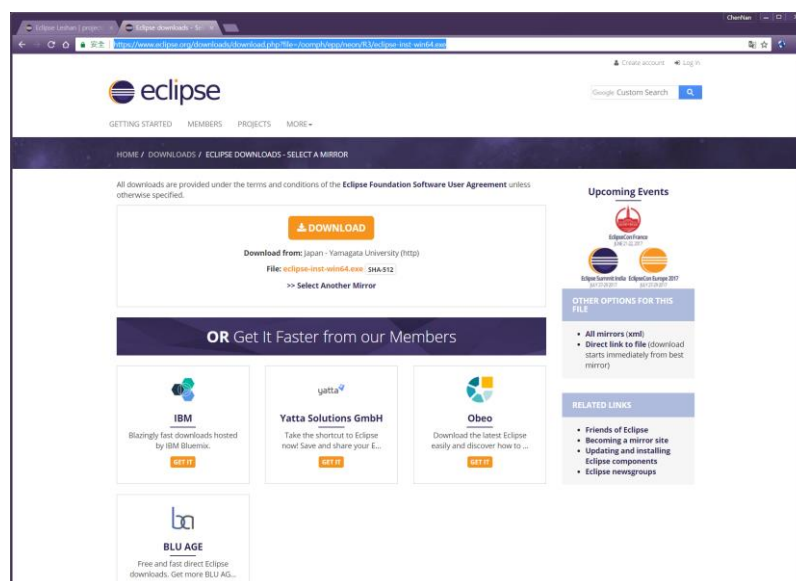


Figure 6 Eclipse home page

3. Install Eclipse IDE on windows by double click the installer.

3.2. Import project

1. Import the lora_ipe project that you download from IoT Ocean homepage to Eclipse.

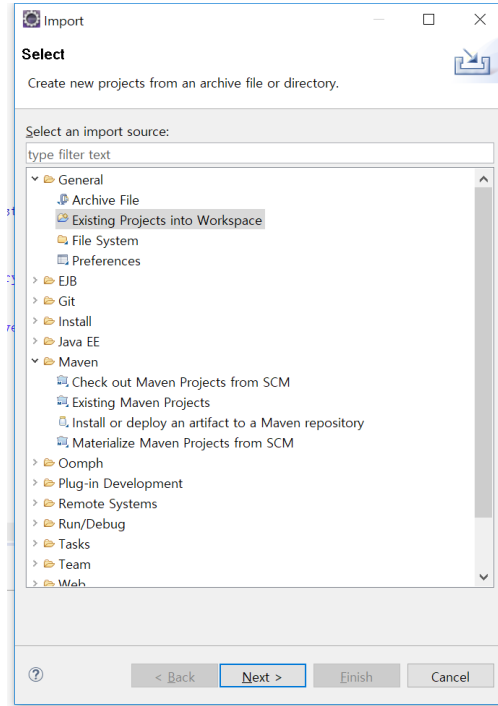


Figure 7 Import ipe project

다운로드 받은 프로젝트가 있는 폴더를 찾아서 import 후 Eclipse IDE의 Project Explorer를 통해 소스파일 확인이 가능함.

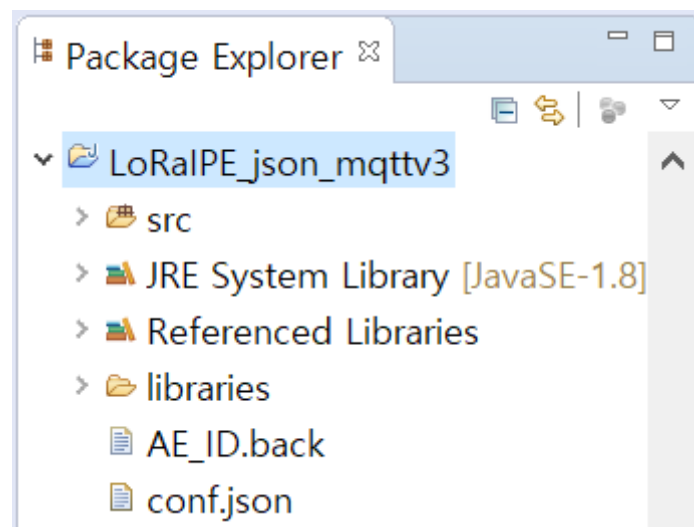


Figure 8 Eclipse Project Explorer

3.3. Execution

1. Export the "lora_ipe" as runnable jar file named "<name>.jar" then copy "<name>.jar" file and "conf.json"(In project) file to target device.

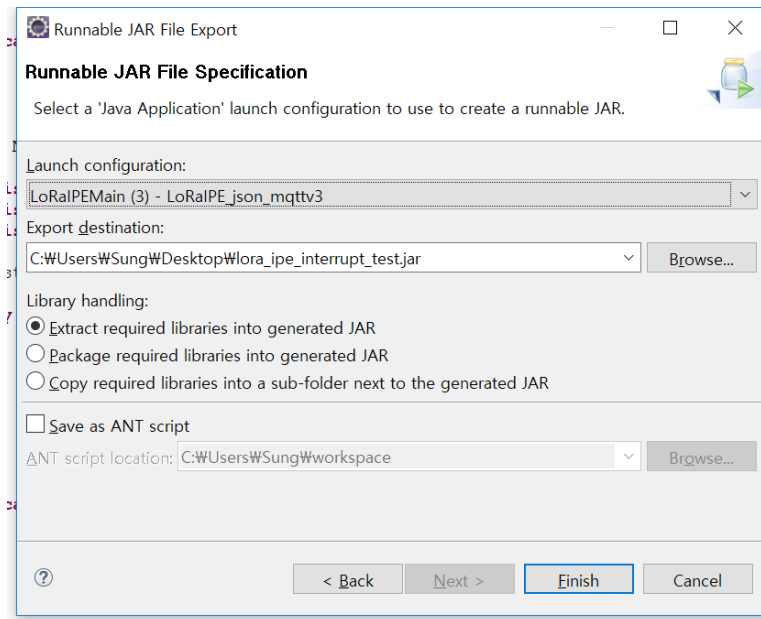


Figure 9 Export execution jar file

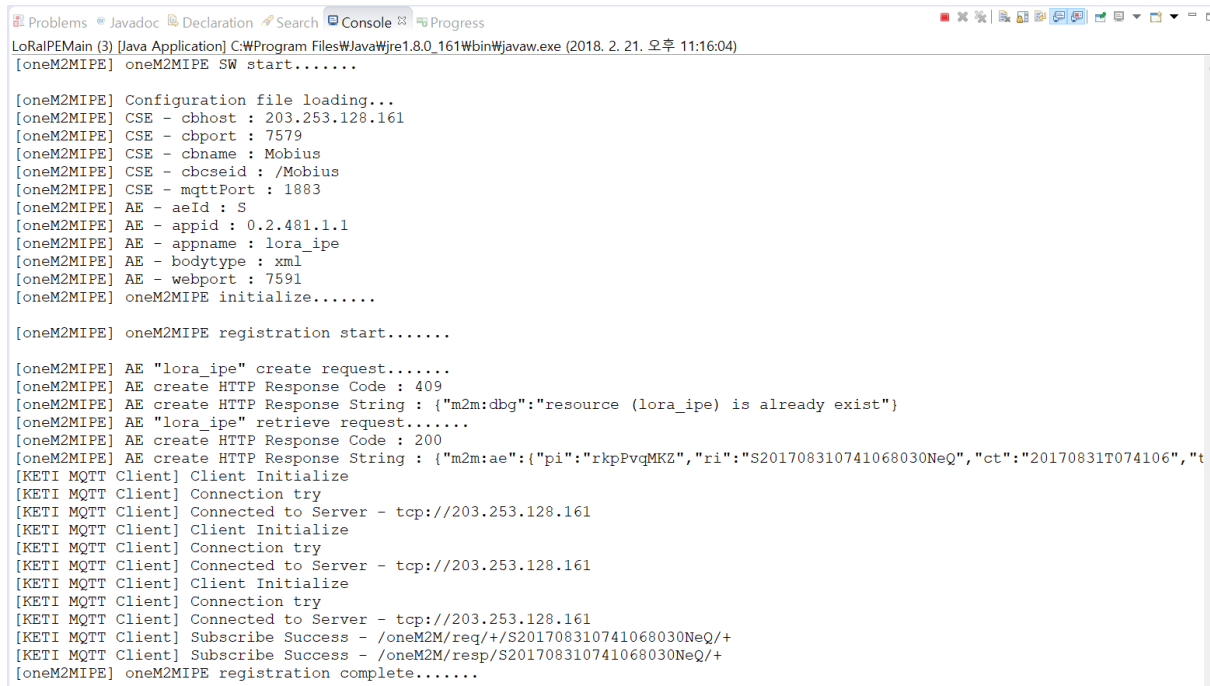
2. Copy the webapp folder to target device jar file path from "lora_ipe" project.
3. Configure the conf.json

```

1 {
2     "cse": {
3         "cbhost": "203.253.128.161",
4         "cbport": "7579",
5         "cbname": "Mobius",
6         "cbcseid": "/Mobius",
7         "mqttport": "1883"
8     },
9     "ae": {
10        "aeid": "S",
11        "appid": "0.2.481.1.1",
12        "appname": "lora_ipe",
13        "bodytype": "xml",
14        "webport": "7591"
15    }
16 }
17 }
```

Figure 10 Configure the tas_conf.xml and port.conf file

4. Run Kerlink(LoRa) IPE.



```

[oneM2MIPE] oneM2MIPE SW start.....

[oneM2MIPE] Configuration file loading...
[oneM2MIPE] CSE - cbhost : 203.253.128.161
[oneM2MIPE] CSE - cbport : 7579
[oneM2MIPE] CSE - cbname : Mobius
[oneM2MIPE] CSE - cbcseid : /Mobius
[oneM2MIPE] CSE - mqttPort : 1883
[oneM2MIPE] AE - aeid : S
[oneM2MIPE] AE - appid : 0.2.481.1.1
[oneM2MIPE] AE - appname : lora_ipe
[oneM2MIPE] AE - bodytype : xml
[oneM2MIPE] AE - webport : 7591
[oneM2MIPE] oneM2MIPE initialize.....

[oneM2MIPE] oneM2MIPE registration start.....

[oneM2MIPE] AE "lora_ipe" create request.....
[oneM2MIPE] AE create HTTP Response Code : 409
[oneM2MIPE] AE create HTTP Response String : {"m2m:dbg":"resource (lora_ipe) is already exist"}
[oneM2MIPE] AE "lora_ipe" retrieve request.....
[oneM2MIPE] AE create HTTP Response Code : 200
[oneM2MIPE] AE create HTTP Response String : {"m2m:ae":{"pi":"rkpPvqMKZ","ri":"S201708310741068030NeQ","ct":"20170831T074106","t
[KETI MQTT Client] Client Initialize
[KETI MQTT Client] Connection try
[KETI MQTT Client] Connected to Server - tcp://203.253.128.161
[KETI MQTT Client] Client Initialize
[KETI MQTT Client] Connection try
[KETI MQTT Client] Connected to Server - tcp://203.253.128.161
[KETI MQTT Client] Client Initialize
[KETI MQTT Client] Connection try
[KETI MQTT Client] Connected to Server - tcp://203.253.128.161
[KETI MQTT Client] Subscribe Success - /oneM2M/req/+/S201708310741068030NeQ/+
[KETI MQTT Client] Subscribe Success - /oneM2M/resp/S201708310741068030NeQ/+
[oneM2MIPE] oneM2MIPE registration complete.....

```

Figure 11 Kerlink(LoRa) IPE execution

4. Test

4.1. Kerlink(LoRa) IPE test

1. Run the Resource Tree Viewer that can download it from IoT Ocean Site.

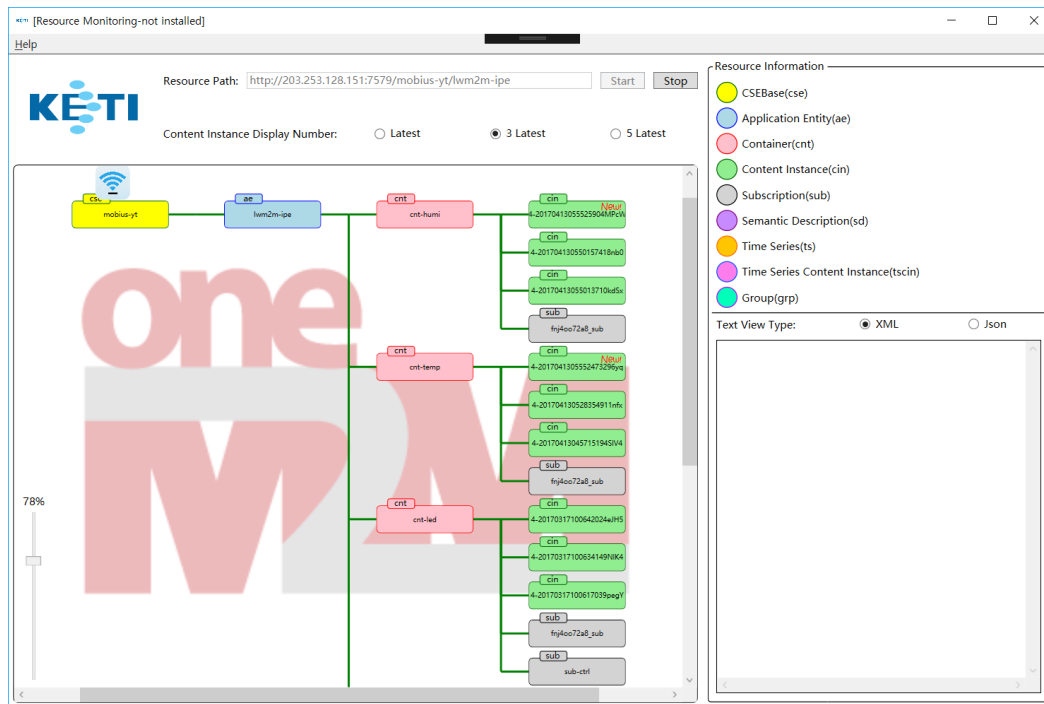


Figure 12 Resource Tree Viewer

Interworking Proxy Entity는 기본적으로 LoRa 표준에서 제시하는 App과 Device를 oneM2M 구조로 매핑하기 위하여 A-EUI(Application ID)를 oneM2M의 AE(Application Entity)로, D-EUI(Device ID)를 oneM2M의 container로 생성하며 커링크 게이트웨이로부터 수신된 LoRa 디바이스의 데이터를 저장하기 위하여 Uplink 패킷용 sub container와 제어를 위한 Downlink 패킷용 sub container를 D-EUI container 하부에 생성함

Downlink 패킷용 sub container의 하부에 subscription 생성으로 외부의 제어 요청 수신 시 해당 제어 메시지를 커링크 게이트웨이로 전달이 가능한 구조를 가짐

그림 13과 같이 리소스 트리 뷰어를 활용하여 Resource Path에 LoRa A-EUI를 입력하여 하위의 D-EUI 및 해당 디바이스에 대한 실시간 단말 데이터 확인 및 제어가 가능함

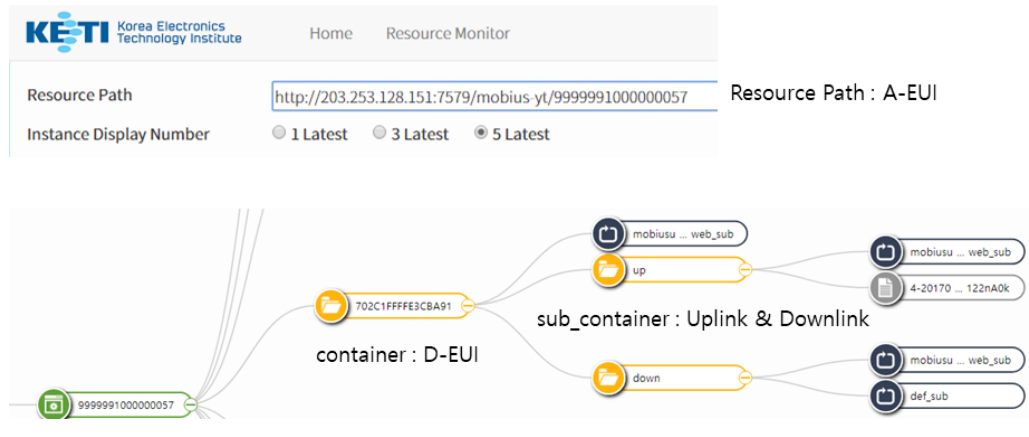


Figure 13 Kerlink(LoRa) IPE 리소스 구조