



Software Version: 2.0.0

Installation Guide v2.0.0

Document Release Date: July 2017

Software Release Date: July 2017



저작권 및 면책조항

본 매뉴얼은 기술상의 부정확함 또는 글자상의 오류를 포함하고 있으며, 저자는 여기에 관한 어떠한 책임도 지지 않습니다.

본 매뉴얼의 내용은 정기적으로 변경 또는 추가 될 수 있으며, 그에 따른 변경본은 본 매뉴얼에 “New Edition”의 형태로 순차적으로 추가될 것입니다. 사전 공지 없이 언제든지 본 매뉴얼에 기재된 제품 및 프로그램을 변경 또는 수정할 수 있습니다.

[Mobius 서버 플랫폼](#) 소스코드는 하기의 라이선스 정책에 따라 배포됩니다.

- OCEAN 기반 오픈 소스코드는 3-clause BSD-style 라이센스를 기반으로 배포한다. 3-clause BSD-style 라이센스에 따라, 소스코드 파일에 저작권 헤더를 유지하는 동안 OCEAN 기반 오픈 소스 코드는 상업용 또는 비상업용 시스템에 자유롭게 사용할 수 있다.
- OCEAN 기반 오픈 소스 라이센스는 개발된 소스코드를 다른 사람과 공유하는 것을 요구하지는 않는다. OCEAN기반 오픈 소스를 활용하여 개발한 소스코드는 개발자의 것이고 누구와 공유할 의무는 없다.
- 누구나 OCEAN의 오픈 소스 발전에 공헌할 수 있다. 그러나, 개발된 소스코드가 OCEAN에 공헌되기를 원한다면 이 코드는 OCEAN의 라이센스 정책을 따르도록 요구된다.

Copyright (c) 2017, 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.

목 차

1.	Mobius 소개	5
1.1.	Mobius 란?	5
1.2.	Mobius 플랫폼	6
1.2.1.	<i>Mobius</i> 서버 플랫폼 개요	6
1.2.2.	<i>Mobius</i> 서버 플랫폼 주요 기능	7
1.2.3.	<i>Mobius</i> 서버 플랫폼 구성	7
1.2.4.	<i>Mobius</i> 서버 플랫폼 S/W 아키텍처	8
1.2.5.	<i>Mobius</i> 서버 디렉토리 구조	9
2.	Mobius 서버 플랫폼 설치	13
2.1.	설치 개요	13
2.2.	Mobius 구동환경 구축 (Linux)	14
2.2.1.	<i>MySQL</i> 설치	14
2.2.2.	<i>MQTT</i> 서버 설치	17
2.2.3.	<i>Node JS</i> 설치	20
2.3.	Mobius 설치	22
2.3.1.	다운로드 및 모듈 설치	22
2.3.2.	<i>mobiusdb</i> 데이터베이스 생성(Linux)	24
3.	Mobius 서버 플랫폼 구동	25
3.1.	환경설정	26
3.2.	테스트	27
3.2.1.	리소스뷰어(<i>oneM2M Browser</i>) 활용	27
3.2.2.	<i>Postman</i> 활용	29

1. Mobius 소개

1.1. Mobius란?

Mobius 플랫폼은 oneM2M 국제 표준을 기반으로 IoT(Internet of Things) 서비스 제공을 위해 다양한 IoT Device 정보를 관리하고, 이들 IoT Device의 접근 제어, 인증, 사용자 관리, 복수의 IoT 서비스 조합을 제공하여 어플리케이션을 통해 서비스하기 위한 플랫폼이다.

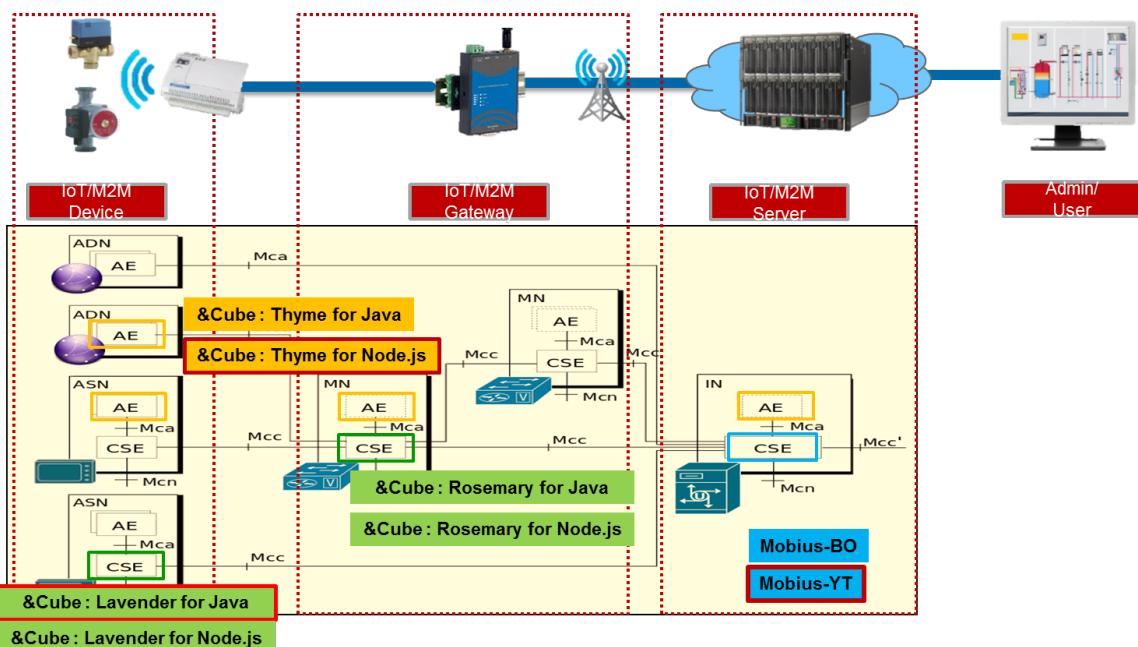


Figure 1 oneM2M 기반 Mobius 플랫폼 개념도

Mobius 플랫폼은 IoT 디바이스와 연동하기 위해 다음 그림과 같이 REST API (http, mqtt, coap, websocket)를 통해 연동된다.

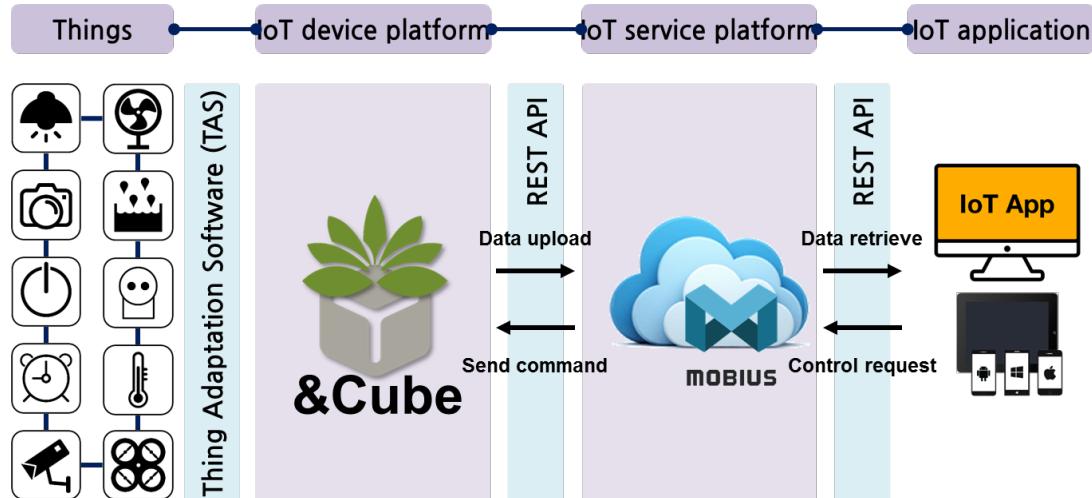


Figure 2 Mobius 와 IoT 디바이스 연동

Mobius 플랫폼은 리소스를 생성하면 자동으로 해당 리소스를 접근할 수 있는 API가 자동으로 생성되며 이를 위한 인터페이스로 REST API인 OPEN API를 제공한다.

1.2. Mobius 플랫폼

1.2.1. Mobius 서버 플랫폼 개요

Mobius Platform은 물리적 기반인 통신 수단을 사용하여 인터넷에 연결된 사물들이 원활하게 작동하도록 관리하고 사물들의 데이터를 저장하는 서버 소프트웨어 플랫폼으로 일종의 미들웨어라고 말할 수 있다. Mobius Server Platform은 디바이스와 어플리케이션의 통신 연결을 쉽고 편리하게 하며, 개방형 개발 환경에서 누구든지 IoT 서비스를 만들어서 사용할 수 있는 생태계를 구축할 수 있도록 지원한다.

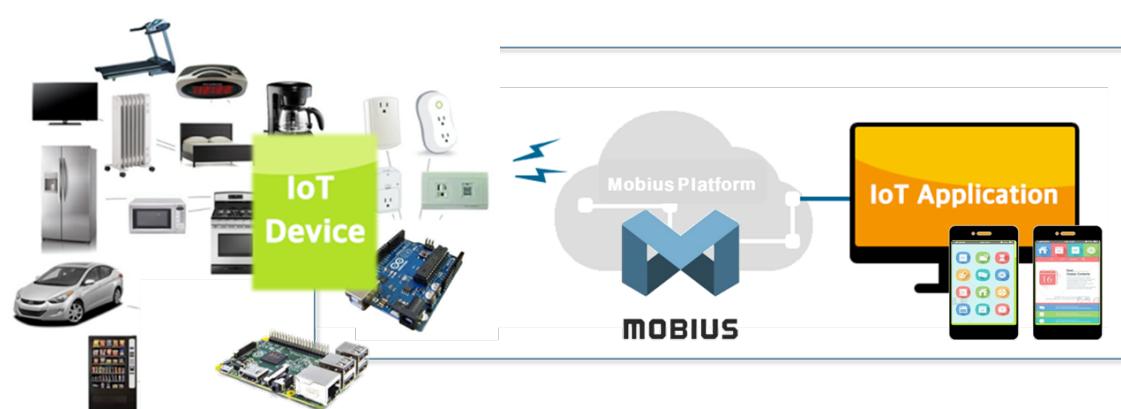


Figure 3 Mobius Platform 개요

Mobius는 node js 를 사용하여 JavaScript 언어로 개발된 Mobius 서버 플랫폼으로 국제 표준인 oneM2M 표준에 따라 개발되었다. Mobius 개발은 node js의 express framework 을 사용하지 않고 express 모듈을 사용하여 개발되었고, http, mqtt, express 등 다양한 node js 모듈을 사용하고 있다.

Mobius Platform은 oneM2M 표준을 준수하여 Http, MQTT, COAP, WebSocket 프로토콜과 Open API를 제공한다. oneM2M 표준에 따라 모비우스의 데이터 구조는 계층적인 리소스 구조를 가지고 있으며, 기본적으로 REST API 방식을 제공합니다. Mobius Platform은 Http Open API를 위한 Http 서버와 MQTT 지원을 위한 MQTT 서버, COAP을 위한 COAP 서버, WebSocket 지원을 위한 WebSocket 서버로 구성되어 있으며, 리소스 저장을 위한 DB는 MySQL DBMS를 사용한다.

1.2.2. *Mobius* 서버 플랫폼 주요 기능

Mobius Platform은 디바이스와 어플리케이션을 연결하는 중간 매개체로써 디바이스는 서버 플랫폼으로 데이터를 전송하고, 플랫폼은 데이터를 저장한다. 어플리케이션은 Open API를 통해 플랫폼에 저장된 데이터를 조회하고 플랫폼으로 제어 요청을 전송한다. 플랫폼은 설정된 디바이스에 따라 어플리케이션의 제어요청을 다시 디바이스로 전달한다.

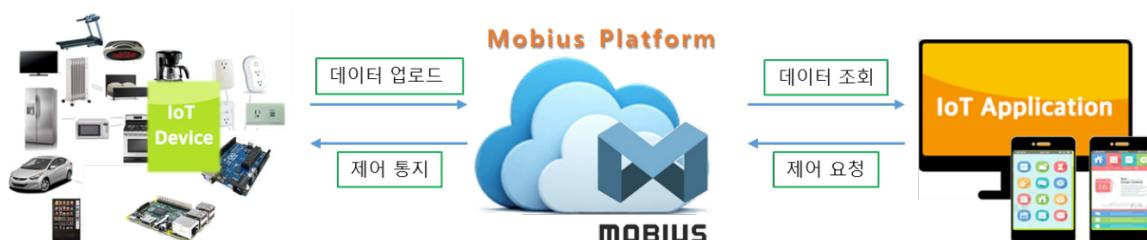


Figure 4 Mobius 서버 플랫폼 주요 기능

1.2.3. *Mobius* 서버 플랫폼 구성

Mobius Platform은 디바이스와 어플리케이션을 연결하며, 디바이스와의 연결을 위해 HTTP, CoAP, MQTT, WebSocket 프로토콜을 지원하며, 리소스 저장을 위한 DB는 MySQL을 사용하여 구성된다

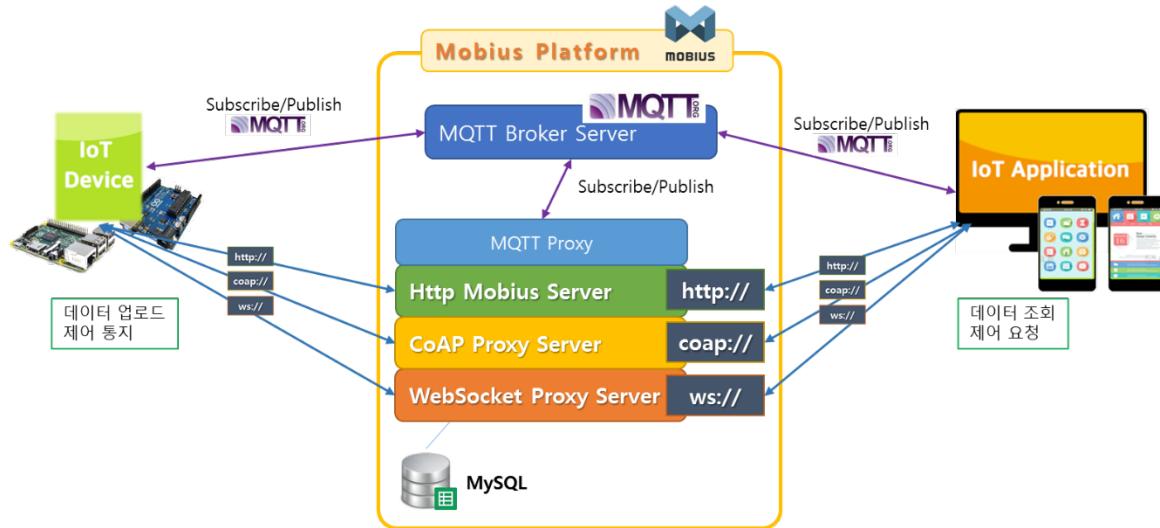


Figure 5 Mobius 서버 플랫폼 시스템 구성

1.2.4. Mobius 서버 플랫폼 S/W 아키텍처

Mobius는 Node JS를 기반으로 JavaScript 언어로 개발되었다. DB는 MySQL을 지원하고 있으며 통신 프로토콜은 HTTP, MQTT, CoAP, WebSocket을 지원한다. 아래는 Mobius의 소프트웨어 아키텍처를 나타낸다.

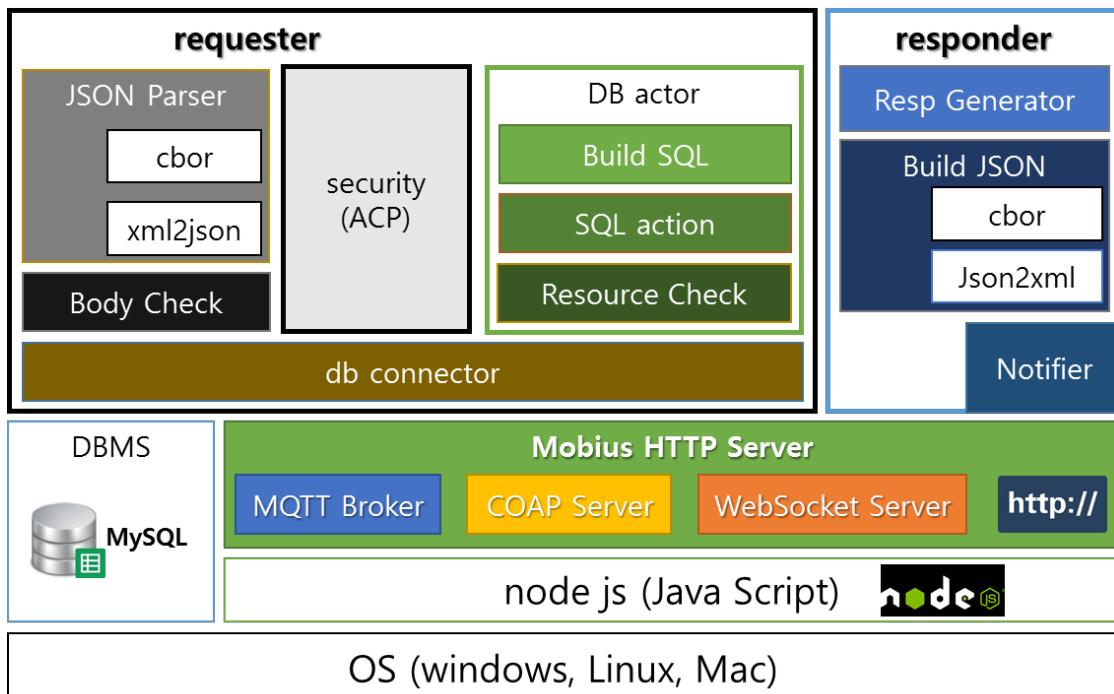


Figure 6 Mobius S/W 아키텍처

위의 Mobius SW 아키텍처는 컴포넌트 개념의 기능 아키텍처를 나타낸다. Mobius 플랫

폼은 기본적으로 http 서버를 중심으로 MQTT Broker, COAP Server, WebSocket Server가 구축된다. 위의 그림처럼 MQTT의 경우 내부적으로 MQTT → HTTP로 변환하는 proxy를 구현하여 MQTT를 지원하고 있고 COAP과 WebSocket도 같은 구조로 구현되어 있다.

Mobius는 requester와 responder 크게 두부분으로 구성되고 requester는 DB 액세스를 위한 컴포넌트를 포함하고 있다. 모든 HTTP request 요청은 requestor 컴포넌트로 전달되고 parser – actor를 거쳐 실제 데이터베이스에 데이터를 액세스하는 sql문을 만들고 db connector를 통해 데이터를 생성하거나 조회, 검색을 수행한다. DB 데이터 액세스 결과가 나오면 request에 대한 응답으로 xml, json, cbor의 데이터 포맷을 결정하고 responder에 의해 처리된다.

Mobius는 윈도우 파일 탐색기와 같은 트리 형태의 리소스 체계를 가지고 이를 지원하고 있다. 리소스 체계는 파일 탐색기의 폴더를 접근하는 것처럼 각 Resource는 Resource 를 접근할 수 있는 URI와 그대로 매핑된다. 예를 들어 IN-CSE 밑에, AE 밑에, container 밑의 데이터를 접근하는 URL은 아래와 같다.

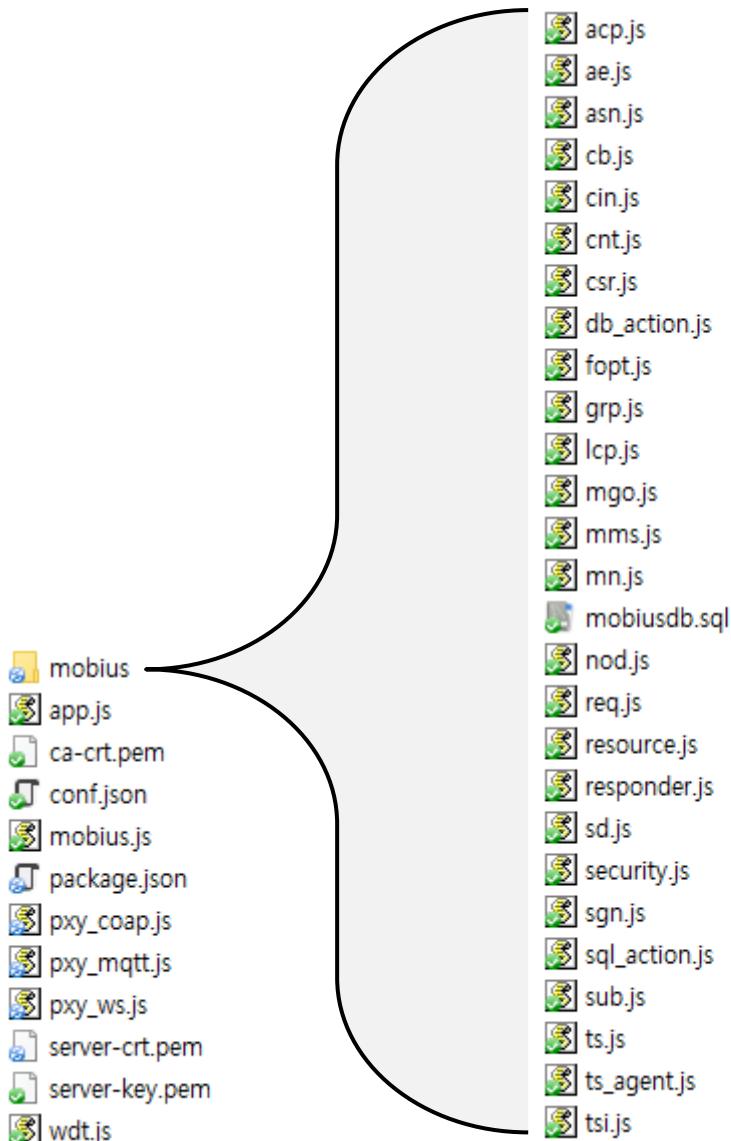
http://hostname:port/IN-CSE_name/AE_name/container_name/contentInstance_name

이처럼 모든 리소스는 위와 같은 방법으로 접근할 수 있으며 POST, GET, PUT, DELETE http method를 통해 생성, 조회, 업데이트, 삭제 동작을 수행할 수 있다. (OCEAN의 API 문서 참조)

Container 밑에 container를 생성할 수 있고 현재 URL의 깊이는 제한하고 있지 않다.

1.2.5. *Mobius* 서버 디렉토리 구조

다음 그림은 Mobius 플랫폼의 디렉토리 구조를 나타낸다.



Mobius 폴더 구조는 크게 mobius 폴더와 mobius.js, app.js, pxy_mqtt.js, pxy_coap.js, pxy_ws.js, wdt.js, conf.json, package.json, openssl key files로 구성된다. mobius 폴더에는 다시 여러 개의 소스 파일로 구성되어 있고 이중에서 특정 역할을 수행하는 db_action.js, fopt.js, resource.js, responder.js, security.js, sgn.js, sql_action.js, ts_agent.js 소스 파일과 단순 리소스 관련 기능을 수행하는 소스파일로 구성되어 있다. 각 파일에 대한 구체적 기능 및 역할은 아래 표와 같다.

소스 파일	기능 및 역할
mobius.js	mobius를 실행하는 파일로써 Mobius 구동을 위한 설정 값들을 수정할 수 있다. 이 파일을 통해 Mobius의 메인 코드를 로드한다.
app.js	Mobius의 main 역할을 수행하는 파일로 Mobius로 들어오는 모든 패킷은 이 파일에서 제일 처음 처리한다. 이 파일에서 http 서버를 실행하고 Listen 하고 대기 한다. 이후 패킷이 들어오면 패킷의 타겟 url을 파싱하고 패킷의 body를 분석하여 정상적인 패킷인지 아닌지를 판단하고, 정상적인 패킷으로 판단되면 해당 동작을 수행하는 resource.js로 전달한다. 오류 패킷일 경우 responder.js를 통해 오류를 리턴한다. 이 파일에서 http 서버는 서버가 실행될 때 서버 성능을 개선하기 위해 클러스터링을 통해 서버를 실행하도록 구현한 것이다.
pxy_mqtt.js	이 파일은 mqtt to http binding 하는 파일로써 mqtt 관련 데이터를 처리하는 파일이다. 시작될 때 mobius 정보를 가져와서 oneM2M mqtt topic을 만들고 subscribe한다. 이후 mqtt로 데이터가 수신되면 이를 http 패킷으로 만들어서 Mobius로 전달하고 response을 받으면 이를 다시 mqtt 패킷으로 만들어 mqtt response 한다.
pxy_coap.js	이 파일은 coap to http binding 하는 파일로써 coap 데이터를 처리하는 파일이다. 시작될 때 coap 서버를 생성하고 이후 coap 데이터가 수신되면 이를 http 패킷으로 만들어서 Mobius로 전달하고 response을 받으면 이를 다시 coap 패킷으로 만들어 coap으로 response 한다.
pxy_ws.js	이 파일은 websocket to http binding 하는 파일로써 websocket 데이터를 처리하는 파일이다. 시작될 때 websocket 서버를 생성하고 이후 websocket 데이터가 수신되면 이를 http 패킷으로 변환해서 Mobius로 전달하고 Mobius로부터 response을 받으면 이를 다시 websocket 패킷으로 만들어 websocket으로 response 한다.
wdt.js	모비우스 자체 watchdogtimer로 구축한 것이고, 모비우스 자체 타이머로 사용된다.

mobius/resource.js	이 파일은 mobius가 지원하는 모든 리소스에 대해 create, retrieve, update, delete, notify, discovery를 수행하는 코어 파일이다. app.js에서 검증된 url, body에 대해 해당되는 method에 따라 내부적으로 처리하는 포맷으로 변환하고 실제 데이터베이스 연동작업을 수행한다. 데이터 베이스 동작은 db_action.js와 sql_action.js를 통해 수행된다.
mobius/responder.js	이 파일은 요청받은 request 패킷에 대해 app.js, resource.js 에 의해 처리된 결과를 response 하는 역할을 수행한다. response 할 때 응답할 데이터를 json, xml 포맷에 맞게 변형한 후 response 한다.
mobius/db_action.js	실제 데이터베이스와 연결하고 데이터베이스에 접근하여 결과를 리턴하는 함수를 포함하는 파일이다.
mobius/sql_action.js	이 파일은 데이터베이스를 Mobius 동작에 필요한 데이터를 얻기위한 sql문을 만들고 이를 db_action.js의 함수를 통해 데이터 베이스로부터 데이터를 리턴받아 전달하는 역할을 수행한다.
mobius/sgn.js	이 파일은 타겟 리소스에 subscription이 차일드로 존재하는지 체크하고 subscription이 존재하면 해당 이벤트를 체크하고 subscription의 notificationUri attribute 정보를 바탕으로 request 메시지를 만들어 전송하는 (notification 전송) 역할을 수행한다.
mobius/security.js	이 파일은 타겟 리소스에 대한 accessControlPolicy를 체킹하여 request 요청에 대해 reject를 할 것인지 허용할 것인지를 판단한다. request 패킷의 X-M2M-Origin 헤더에 포함되어 있는 id 값을 가지고 타겟 리소스에 연결되어 있는 accessControlPolicy 리소스를 확인하여 이 id에 접근 권한이 있는지 확인한다.
mobius/fopt.js	group 리소스에 대한 fanOutPoint 리소스에 대한 처리를 담당한다. fanOutPoint 리소스는 virtual 리소스로써 group 리소스에 포함된 멤버들 모두에게 요청받은 패킷을 전달하고 그 결과를 모아서 요청자에게 response 한다.
mobius/ts_agent.js	이 파일은 timeSeriesInstance 리소스를 관리하는 역할로써 timeSeriesInstance 중에서 missing 되는 데이터들을 주기적으로 모니터링하여 missing 데이터들을 저장한다.
mobius/acp.js	accessControlPolicy 리소스 파일 – 접근제어 관련 리소스
mobius/ae.js	AE 리소스 파일
mobius/cb.js	CSEBase 리소스 파일 - Mobius 자신의 정보를 가진다.
mobius/cin.js	contentInstance 리소스 파일 – 실제 데이터가 여기에 저장된다.

mobius/cnt.js	container 리소스 파일
mobius/csr.js	remoteCSE 리소스 파일
mobius/grp.js	group 리소스 파일
mobius/lcp.js	locationPolicy 리소스 파일
mobius/mms.js	multimediaSession 리소스 파일
mobius/sd.js	semanticDescriptor 리소스 파일
mobius/sub.js	subscription 리소스 파일
mobius/ts.js	timeSeries 리소스 파일
mobius/tsi.js	timeSeriesInstance 리소스 파일
mobius/req.js	request 리소스 파일 – nonblocking 기능 지원
mobius/nod.js	node 리소스 파일
mobius/mgo.js	mgmtObj 리소스 파일

2. Mobius 서버 플랫폼 설치

2.1. 설치 개요

Mobius Platform은 MySQL DBMS를 사용한다. 먼저 MySQL을 설치하고 MQTT Broker, NodeJS 설치 순으로 설치를 하면 Mobius 서버 구동을 위한 환경이 완성 되고 이후 Mobius 서버를 설치하여 플랫폼 구축을 완료한다.



Figure 7 Mobius 서버 플랫폼 설치 개요

2.2. Mobius 구동환경 구축 (Linux)

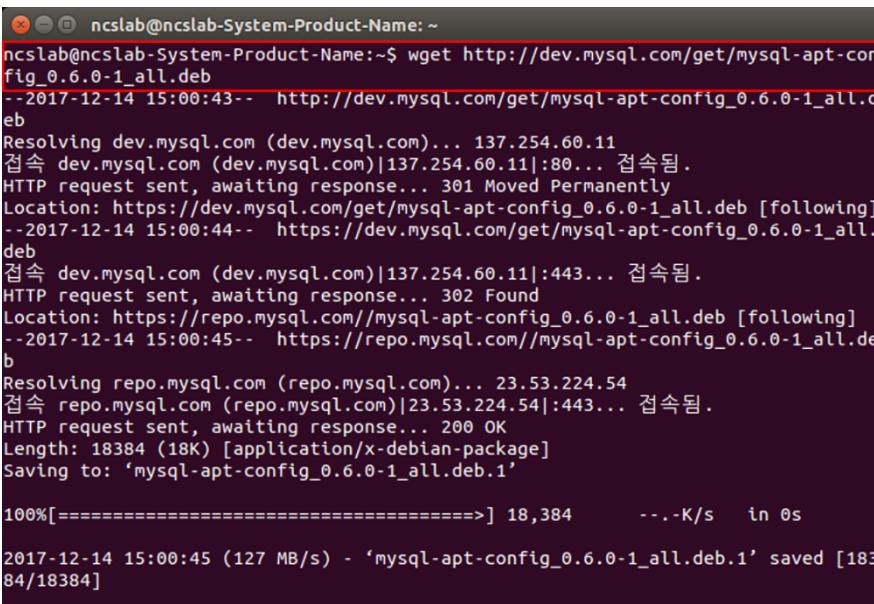
2.2.1. MySQL 설치

Mobius는 데이터 저장 및 관리를 위해 MySQL DB를 기반으로 동작한다. 여기서는 오픈 소스 RDBMS인 MySQL을 다운로드 받고 설치하는 방법에 대해 설명한다. Linux 버전은 14.04버전으로 진행하였다.

<http://www.mysql.com>



The image shows a screenshot of the MySQL.com website. At the top, there's a navigation bar with links for Downloads, Documentation, Developer Zone, Products, Services, Partners, Customers, Why MySQL?, News & Events, and How to Buy. Below the navigation bar, there's a large banner for 'Oracle Enterprise Manager Cloud Control'. The banner displays various performance metrics and graphs, including Availability History, Transactions Activity, Response Time (ms), and Resource Activity. Below the banner, there's a section titled 'Downloads' with a link to 'mysql-apt-config_0.6.0-1_all.deb'.

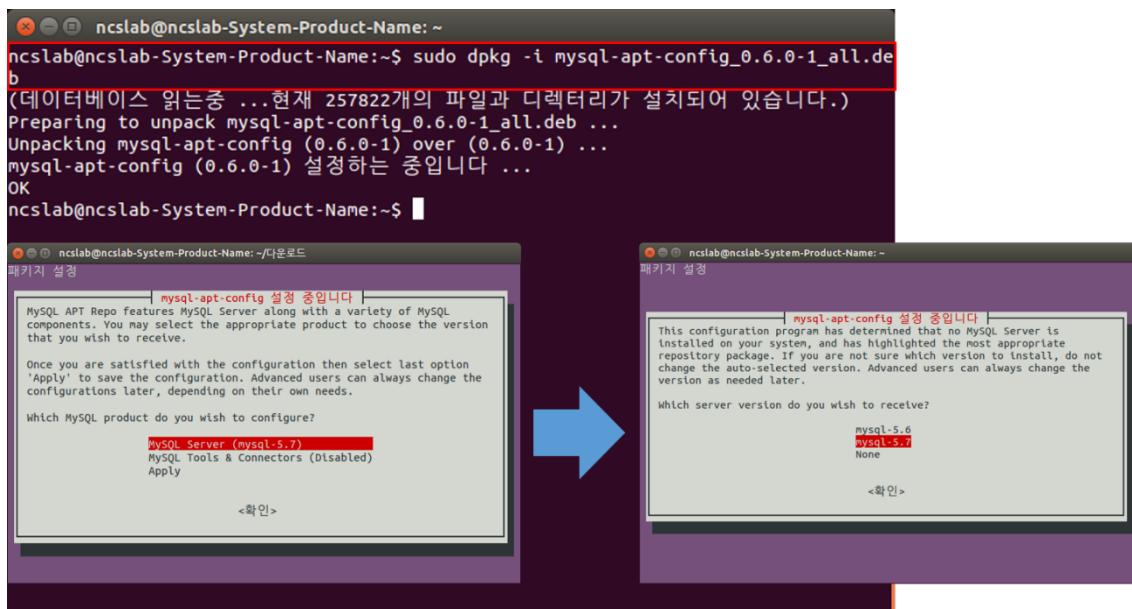


```
ncslab@ncslab-System-Product-Name: ~
ncslab@ncslab-System-Product-Name: ~$ wget http://dev.mysql.com/get/mysql-apt-config_0.6.0-1_all.deb
--2017-12-14 15:00:43-- http://dev.mysql.com/get/mysql-apt-config_0.6.0-1_all.deb
Resolving dev.mysql.com (dev.mysql.com)... 137.254.60.11
접속 dev.mysql.com (dev.mysql.com)|137.254.60.11|:80... 접속됨.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://dev.mysql.com/get/mysql-apt-config_0.6.0-1_all.deb [following]
--2017-12-14 15:00:44-- https://dev.mysql.com/get/mysql-apt-config_0.6.0-1_all.deb
Resolving dev.mysql.com (dev.mysql.com)|137.254.60.11|:443... 접속됨.
HTTP request sent, awaiting response... 302 Found
Location: https://repo.mysql.com/mysql-apt-config_0.6.0-1_all.deb [following]
--2017-12-14 15:00:45-- https://repo.mysql.com/mysql-apt-config_0.6.0-1_all.deb
Resolving repo.mysql.com (repo.mysql.com)... 23.53.224.54
접속 repo.mysql.com (repo.mysql.com)|23.53.224.54|:443... 접속됨.
HTTP request sent, awaiting response... 200 OK
Length: 18384 (18K) [application/x-debian-package]
Saving to: 'mysql-apt-config_0.6.0-1_all.deb.1'

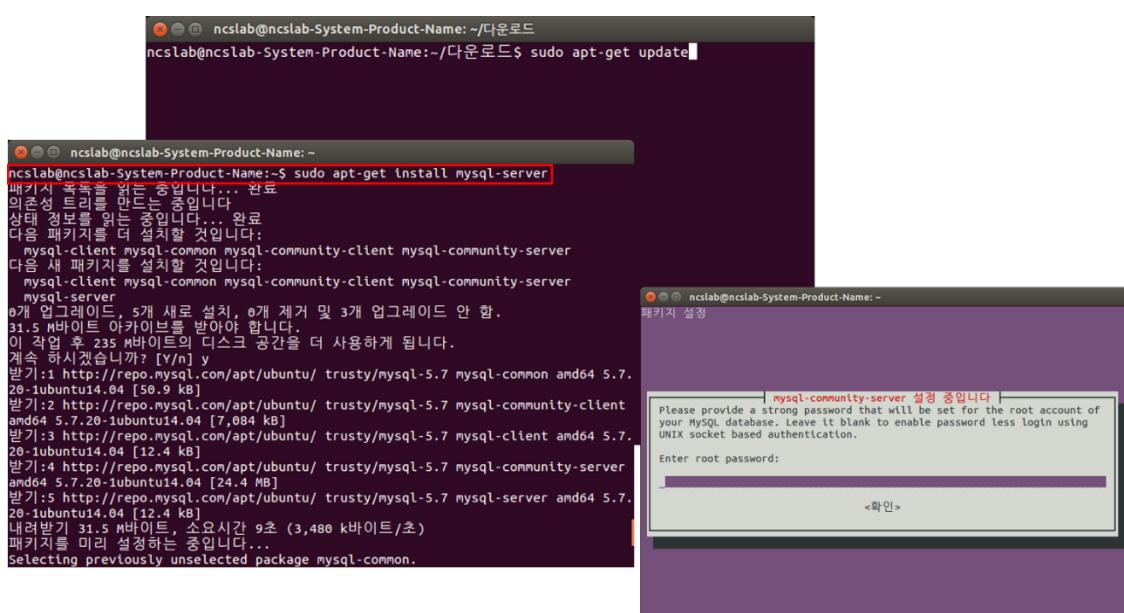
100%[=====] 18,384      --K/s   in 0s

2017-12-14 15:00:45 (127 MB/s) - 'mysql-apt-config_0.6.0-1_all.deb.1' saved [18384/18384]
```

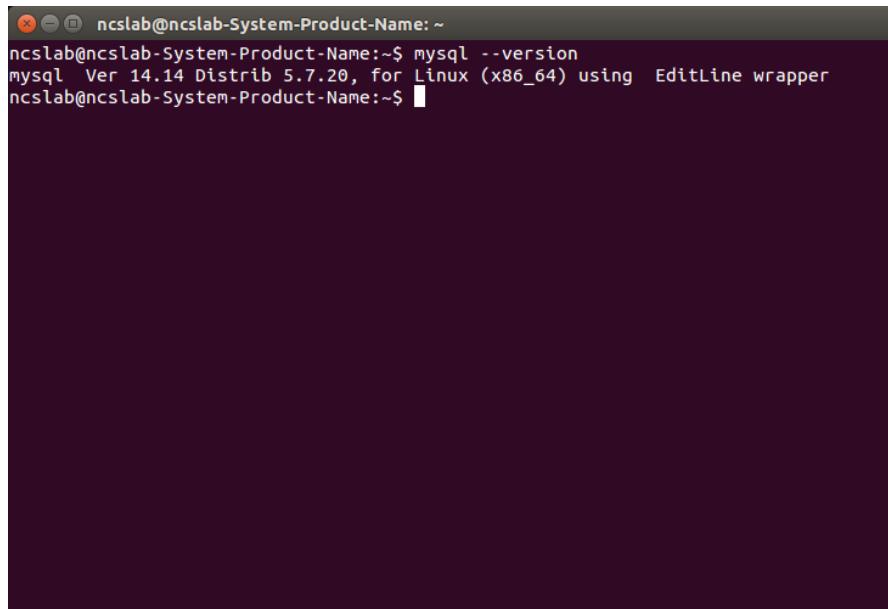
MySQL 5.7 버전은 우분투 버전 16.04 이상은 추가 패키지설치없이 진행되기 때문에 “apt-get install mysql-server”라는 명령어만으로 설치 가능하다. 14.04 버전 우분투 버전은 MySQL 5.7버전 설치를 위해 deb파일을 서버로부터 직접 다운받아 패키지설치리스트에 추가해야 한다. “wget http://dev.mysql.com/get/mysql-apt-config_0.6.0-1_all.deb” 다음과 같은 명령어를 통해 다운로드한다.



다운받은 deb파일을 “dpkg” 명령어를 사용하여 설치합니다. 명령어를 입력하면 설치하려는 MySQL을 선택하는 화면이 나옵니다. MySQL Server(mysql-5.7)를 선택하고 ENTER를 눌러 5.7버전 선택한 후, Apply를 선택하여 ENTER를 누른다.

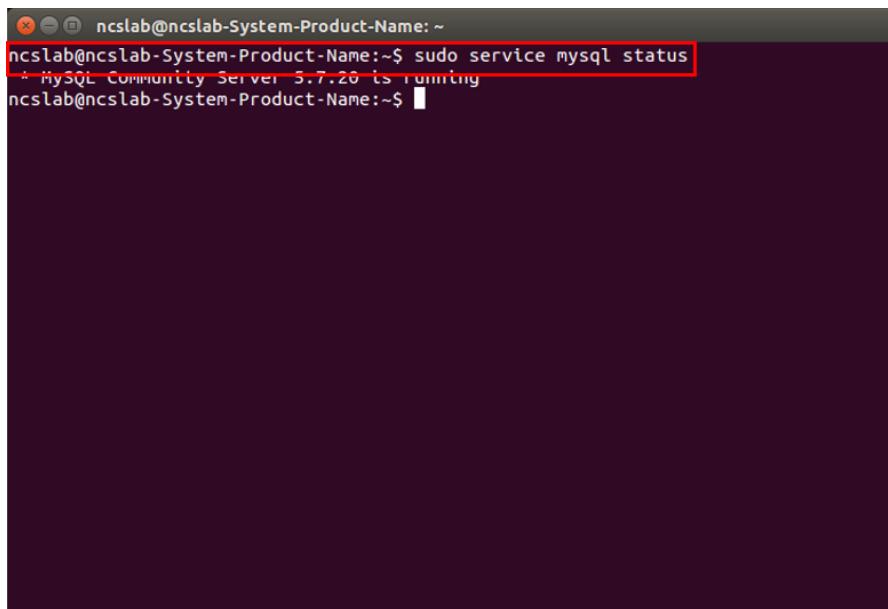


"sudo apt-get update" 명령어를 통해 설치패키지리스트를 최신화한 후 "sudo apt-get install mysql-server" 입력하여 설치를 진행한다. 설치 진행중에 루트 암호를 설정해야하는 화면이 나온다. 이 암호는 추후에 Mobius와 연동을 위해 필요한 암호이기 때문에 꼭 기억 할 수 있는 암호로 설정해야한다.



```
ncslab@ncslab-System-Product-Name: ~
ncslab@ncslab-System-Product-Name:~$ mysql --version
mysql Ver 14.14 Distrib 5.7.20, for Linux (x86_64) using EditLine wrapper
ncslab@ncslab-System-Product-Name:~$
```

"mysql -version" 명령어를 통해 설치된 mysql 버전이 5.7버전인지 확인을 할 수 있다.



```
ncslab@ncslab-System-Product-Name: ~
ncslab@ncslab-System-Product-Name:~$ sudo service mysql status
 * MySQL Community Server 5.7.20 is running
ncslab@ncslab-System-Product-Name:~$
```

mysql 설치가 완료되면 mysql은 자동으로 실행이 된다. 하지만 실행이 정상으로 되고 있는지 확인을 위해 "sudo service mysql status"를 입력하여 확인한다.

2.2.2. MQTT 서버 설치

Mobius Platform은 디바이스와의 통신을 위해 MQTT 프로토콜을 지원하고 있으며 이를 위해서 오픈소스로 제공되는 MQTT broker Mosquitto 서버를 설치한다.

<http://mosquitto.org>

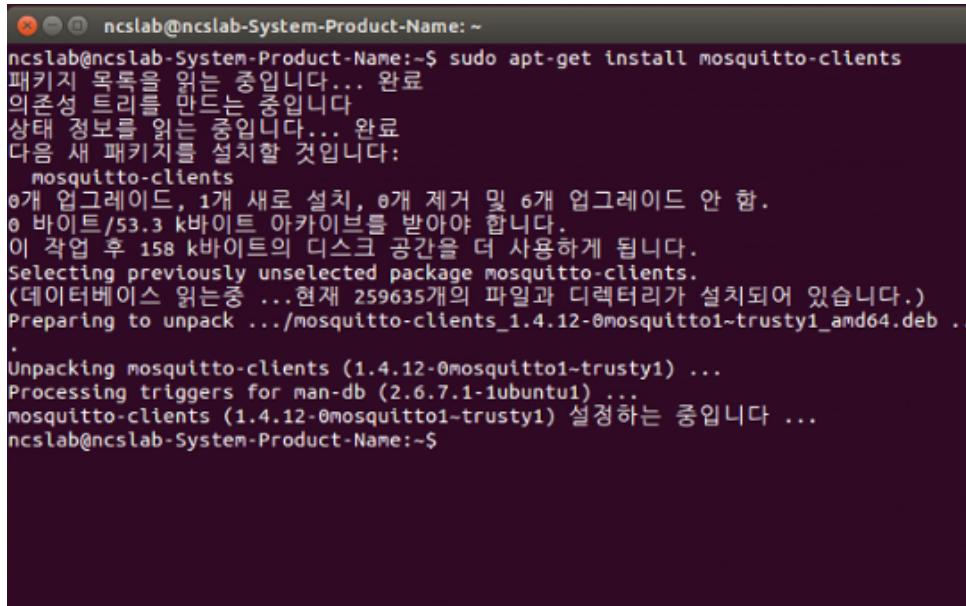
위의 사이트에서 버전을 확인 할 수 있으며 만약 MQTT를 지원하지 않는다면 설치를 하지 않아도 무방하다. 현재 Linux버전에서 지원하는 MQTT 버전으로 설치하면 된다.

The screenshot shows the Mosquitto website's download page. It features a logo with a blue and orange bird icon, the text "Mosquitto An Open Source MQTT v3.1/v3.1.1 Broker", and a "Home" link. Below this, there are sections for "Downloads" and "Source". The "Downloads" section includes a "Donation" button and payment method icons for American Express, MasterCard, Visa, and Direct Debit. The "Source" section lists download links for "mosquitto-1.4.8.tar.gz" (GPG signature), "mosquitto-1.4.8.tar.gz (via Eclipse)", and "Git source code repository (github.com)". It also mentions older downloads at <http://mosquitto.org/files/> and a deprecated javascript/websockets client available at <http://mosquitto.org/s/mosquitto-1.1.js>. The terminal window below shows the command `sudo apt-get install mosquitto` being run in a terminal session, with Korean text indicating the progress of the package installation.

```

ncslab@ncslab-System-Product-Name: ~
ncslab@ncslab-System-Product-Name:~$ sudo apt-get install mosquitto
파키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음 새 패키지를 설치할 것입니다:
mosquitto
0개 업그레이드, 1개 새로 설치, 0개 제거 및 6개 업그레이드 안 함.
0 바이트/130 k바이트 아카이브를 받아야 합니다.
이 작업 후 370 k바이트의 디스크 공간을 더 사용하게 됩니다.
Selecting previously unselected package mosquitto.
(데이터베이스 읽는 중 ...현재 259626개의 파일과 디렉터리가 설치되어 있습니다.)
Preparing to unpack .../mosquitto_1.4.12-0mosquitto1-trusty1_amd64.deb ...
Unpacking mosquitto (1.4.12-0mosquitto1-trusty1) ...
Processing triggers for ureadahead (0.100.0-16) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
mosquitto (1.4.12-0mosquitto1-trusty1) 설정하는 중입니다 ...
mosquitto start/running, process 5700
ncslab@ncslab-System-Product-Name:~$ 

```



```
ncslab@ncslab-System-Product-Name: ~
ncslab@ncslab-System-Product-Name:~$ sudo apt-get install mosquito-clients
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음 새 패키지를 설치할 것입니다:
mosquitto-clients
0개 업그레이드, 1개 새로 설치, 0개 제거 및 6개 업그레이드 안 함.
0 바이트/53.3 kB바이트 아카이브를 받아야 합니다.
이 작업 후 158 kB바이트의 디스크 공간을 더 사용하게 됩니다.
Selecting previously unselected package mosquito-clients.
(데이터베이스 읽는중 ...현재 259635개의 파일과 디렉터리가 설치되어 있습니다.)
Preparing to unpack .../mosquitto-clients_1.4.12-0mosquitto1-trusty1_amd64.deb ..
.
Unpacking mosquito-clients (1.4.12-0mosquitto1-trusty1) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
mosquitto-clients (1.4.12-0mosquitto1-trusty1) 설정하는 중입니다 ...
ncslab@ncslab-System-Product-Name:~$
```

Windows mosquitto는 홈페이지에서 하나만 다운로드하여 설치하면 되지만 Linux에서는 mosquitto와 mosquitto-clients 두 가지를 각각 설치해야 한다. 먼저 mosquitto 서버는 “sudo apt-get install mosquitto” 명령어를 통해서 설치한다. 다음으로 mosquitto client는 “sudo apt-get install mosquitto-clients” 명령어를 통해 설치한다.



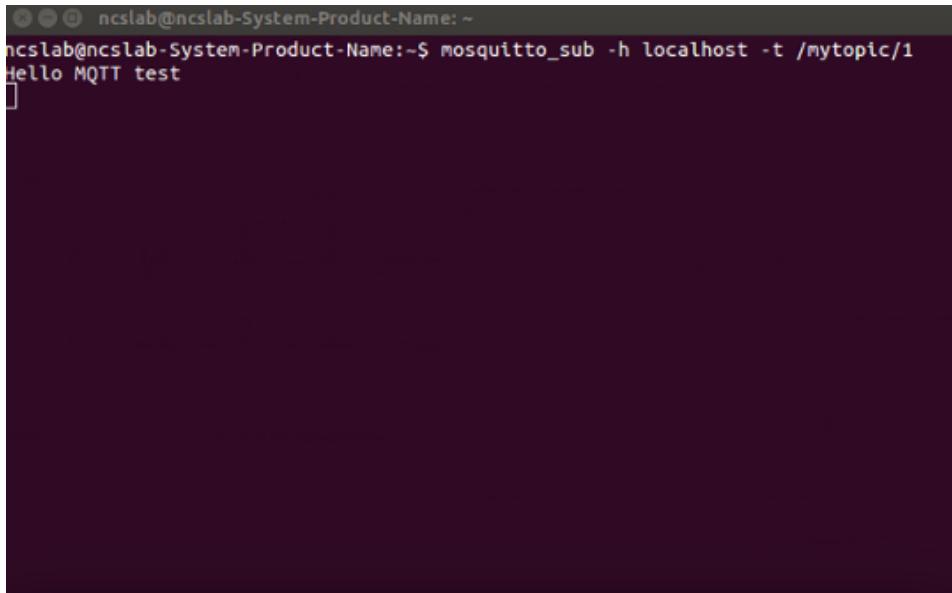
```
ncslab@ncslab-System-Product-Name: ~
ncslab@ncslab-System-Product-Name:~$ mosquito -v
1514971687: mosquito version 1.4.12 (build date Sat, 27 May 2017 21:38:19 +0100)
starting
1514971687: Using default config.
1514971687: Opening ipv4 listen socket on port 1883.
1514971687: Error: Address already in use
ncslab@ncslab-System-Product-Name:~$
```

설치된 mosquito 버전을 “mosquitto -v” 명령어를 통해 확인한다.

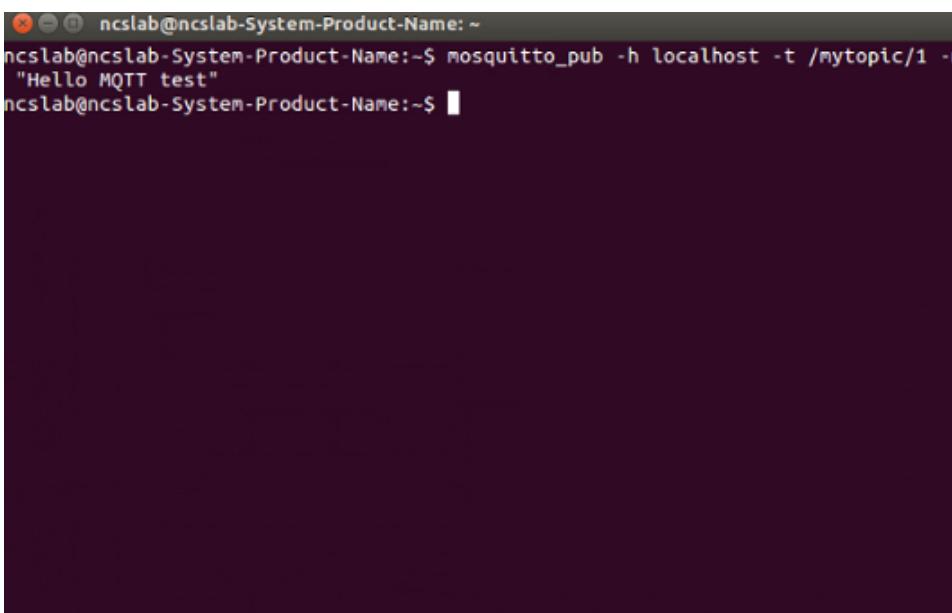
다음 과정을 통해 설치된 mosquitto를 동작테스트를 할 수 있다.

```
mosquitto_sub.exe -h localhost -t /mytopic/1
```

```
mosquitto_pub.exe -h localhost -t /mytopic/1 -m "Hello MQTT test"
```



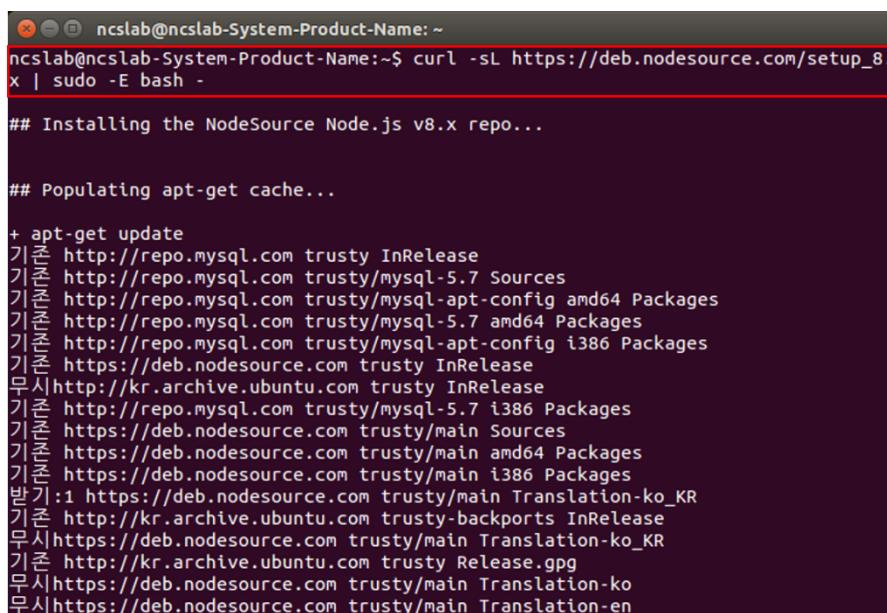
```
ncslab@ncslab-System-Product-Name: ~
ncslab@ncslab-System-Product-Name:~$ mosquitto_sub -h localhost -t /mytopic/1
Hello MQTT test
```



```
ncslab@ncslab-System-Product-Name: ~
ncslab@ncslab-System-Product-Name:~$ mosquitto_pub -h localhost -t /mytopic/1 -m
"Hello MQTT test"
ncslab@ncslab-System-Product-Name:~$
```

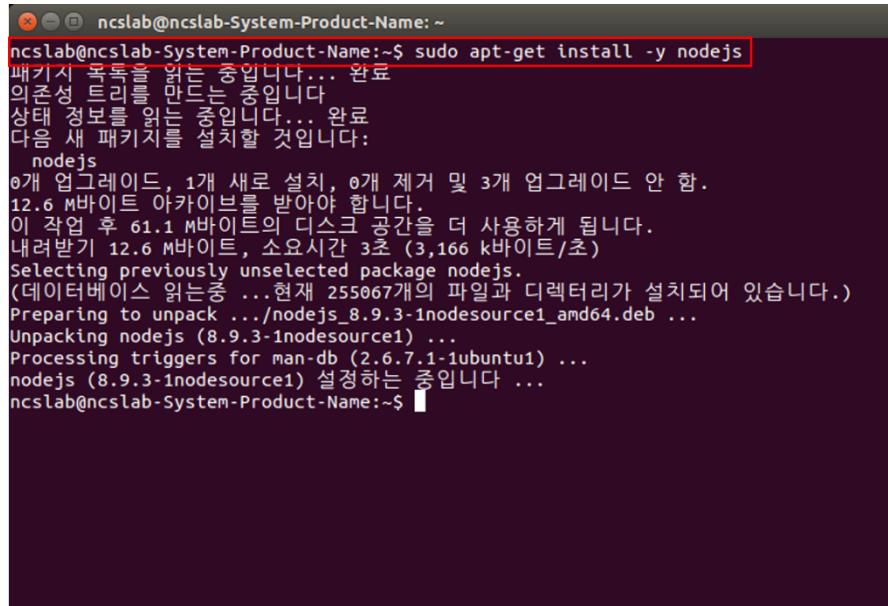
2.2.3. Node JS 설치

Node.js는 고성능의 비동기 IO (Async/Non-blocking IO)를 지원하는 single thread 기반 네트워크 서버이다. 2009년 Ryan Dahl에 의해 개발이 시작되었고 현재 수많은 지원 모듈을 가지고 있는 오픈 소스 프로젝트 중 하나이다. Node.js는 Google Chrome V8 엔진으로 개발되어 있고 Event 기반의 프로그래밍 모델로써 프로그래밍 언어로는 Java script를 사용한다. 근래에 들어서 많은 인터넷 기업들이 Node.js를 도입하고 있다.



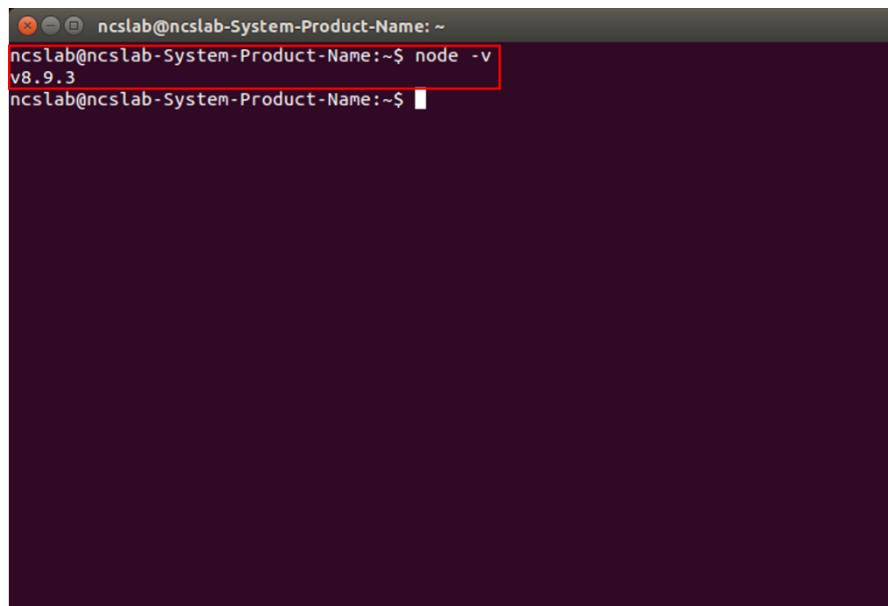
```
ncslab@ncslab-System-Product-Name: ~
ncslab@ncslab-System-Product-Name:~$ curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -
## Installing the NodeSource Node.js v8.x repo...
.
.
## Populating apt-get cache...
+ apt-get update
기존 http://repo.mysql.com trusty InRelease
기존 http://repo.mysql.com trusty/mysql-5.7 Sources
기존 http://repo.mysql.com trusty/mysql-apt-config amd64 Packages
기존 http://repo.mysql.com trusty/mysql-5.7 amd64 Packages
기존 http://repo.mysql.com trusty/mysql-apt-config i386 Packages
기존 https://deb.nodesource.com trusty InRelease
무시http://kr.archive.ubuntu.com trusty InRelease
기존 http://repo.mysql.com trusty/mysql-5.7 i386 Packages
기존 https://deb.nodesource.com trusty/main Sources
기존 https://deb.nodesource.com trusty/main amd64 Packages
기존 https://deb.nodesource.com trusty/main i386 Packages
받기 :1 https://deb.nodesource.com trusty/main Translation-ko_KR
기존 http://kr.archive.ubuntu.com trusty-backports InRelease
무시https://deb.nodesource.com trusty/main Translation-ko_KR
기존 http://kr.archive.ubuntu.com trusty Release.gpg
무시https://deb.nodesource.com trusty/main Translation-ko
무시https://deb.nodesource.com trusty/main Translation-en
```

"sudo apt-get install nodejs" 명령어를 통해 v6.x 버전을 설치하는 것도 가능하지만 여기에서는 v8.x 버전을 설치 할 것이다. v8.x은 패키지에서 바로 설치하는 것이 불가능하기 때문에 데비안과 우분투 운영체제의 바이너리 배포판 저장소 Node source 홈페이지에서 제공 하는 최신 v8 LTS을 "curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -" 명령어를 통해 apt 저장소를 추가한다.



```
ncslab@ncslab-System-Product-Name: ~
ncslab@ncslab-System-Product-Name:~$ sudo apt-get install -y nodejs
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음 새 패키지를 설치할 것입니다:
  nodejs
0개 업그레이드, 1개 새로 설치, 0개 제거 및 3개 업그레이드 안 함.
12.6 M바이트 아카이브를 받아야 합니다.
이 작업 후 61.1 M바이트의 디스크 공간을 더 사용하게 됩니다.
내려받기 12.6 M바이트, 소요시간 3초 (3,166 kB바이트/초)
Selecting previously unselected package nodejs.
(데이터베이스 읽는중 ...현재 255067개의 파일과 디렉터리가 설치되어 있습니다.)
Preparing to unpack .../nodejs_8.9.3-1nodesource1_amd64.deb ...
Unpacking nodejs (8.9.3-1nodesource1) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
nodejs (8.9.3-1nodesource1) 설정하는 중입니다 ...
ncslab@ncslab-System-Product-Name:~$
```

"sudo apt-get install -y nodejs" 명령어를 통해 설치한다.



```
ncslab@ncslab-System-Product-Name: ~
ncslab@ncslab-System-Product-Name:~$ node -v
v8.9.3
ncslab@ncslab-System-Product-Name:~$
```

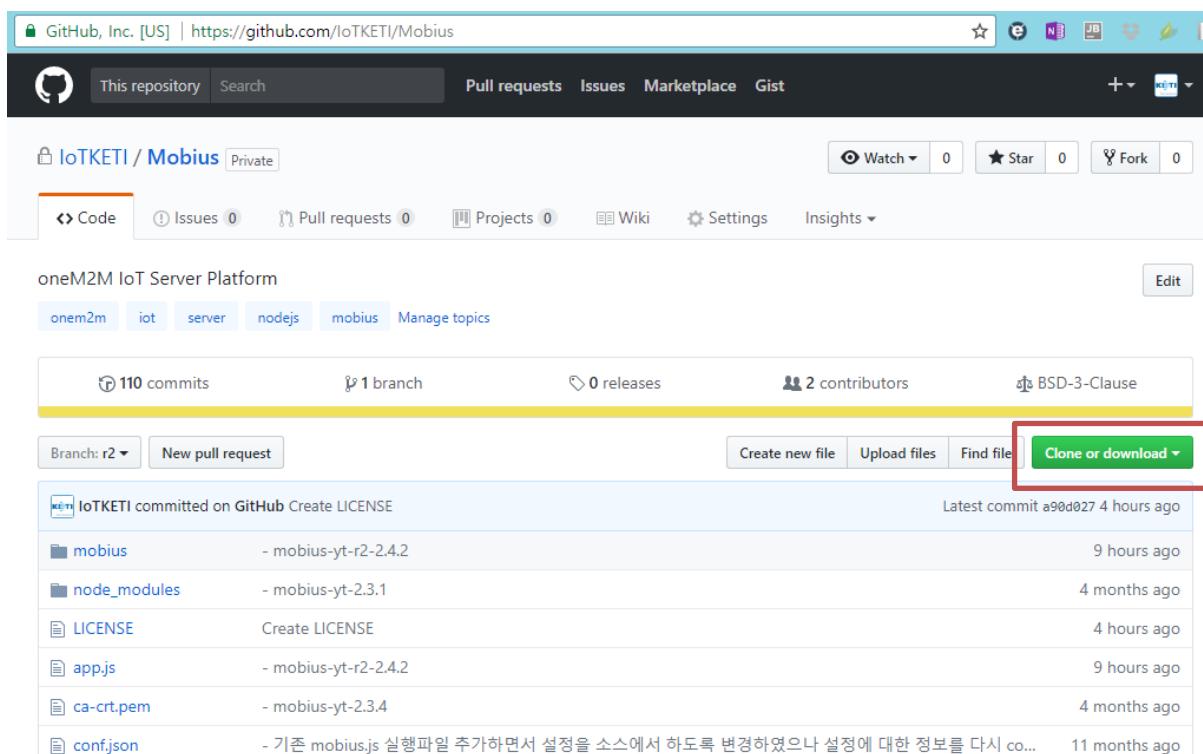
"node -v" 명령어를 통해 설치한 버전을 확인한다.

2.3. Mobius 설치

Mobius Platform은 GitHub를 통해 Mobius 서버 플랫폼 소스를 공개하고 있다. Mobius 서버 플랫폼 압축 파일을 다운로드 받은 후, 제공된 매뉴얼에 따라 Port 정보 등의 Configuration 정보를 설정한다.

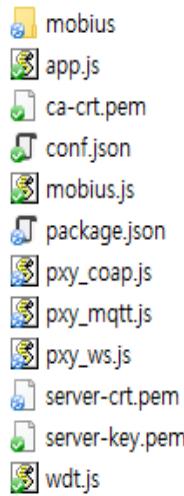
2.3.1. 다운로드 및 모듈 설치

Mobius 설치 파일은 OCEAN Alliance에서 제공하는 GitHub (<https://github.com/IoTKETI/Mobius>) 접속하여 아래와 같이 Mobius 서버 파일을 다운로드 한다.



The screenshot shows the GitHub repository page for 'IoTKETI / Mobius'. At the top, there's a navigation bar with links for 'Pull requests', 'Issues', 'Marketplace', and 'Gist'. Below the navigation bar, the repository name 'IoTKETI / Mobius' is displayed, along with 'Private' status, 'Watch' (0), 'Star' (0), and 'Fork' (0) buttons. The main content area shows repository statistics: 110 commits, 1 branch, 0 releases, and 2 contributors. A yellow bar highlights these stats. Below the stats, there are buttons for 'Create new file', 'Upload files', 'Find file', and a prominent green 'Clone or download' button, which is also highlighted with a red box. The repository's history is listed below, showing commits from 'IoTKETI' and others, including files like 'LICENSE', 'app.js', 'ca-crt.pem', and 'conf.json'.

다운로드한 파일을 적당한 곳에서 압축을 풀면 다음과 같이 node.js 버전의 Mobius 파일들을 볼 수 있다. Node.js는 별도의 컴파일 과정이 필요하지 않으므로 바로 압축을 푸 후 실행할 수 있다. 이전의 MySQL과 Node.js가 제대로 설치가 되어 있다면 Mobius에 사용된 node.js 모듈을 설치하면 별도의 작업없이 실행이 가능하다.



mobius.js가 Mobius의 메인 코드를 실행하는 파일이다. app.js는 대문 역할과 라우팅 역할을 수행하는 메인 코드이다. mobius 디렉토리에 있는 파일들이 실제 패킷을 처리하고 DB 액세스와 응답을 수행하는 코드들이다. 현재는 추가적인 node.js 모듈들이 설치가 되어있지 않아 에러가 날 것이다. 다음과 같이 명령 프롬프트를 실행시키고 npm 툴을 이용해 필요한 모듈을 설치해야 한다. package.json 파일에 필요한 모듈 리스트를 가지고 있다.

```
ncslab@ncslab-System-Product-Name:~/다운로드/Mobius-Mobius-2.0.0
ncslab@ncslab-System-Product-Name:~/다운로드/Mobius-Mobius-2.0.0$ npm install
npm [WARN] deprecated crypto@0.0.3: This package is no longer supported. It's now
a built-in Node module. If you've depended on crypto, you should switch to the o
ne that's built-in.

> websocket@1.0.25 install /home/ncslab/다운로드/Mobius-Mobius-2.0.0/node_modules/
s/websocket
> (node-gyp rebuild 2> builderror.log) || (exit 0)

make: Entering directory `/home/ncslab/다운로드/Mobius-Mobius-2.0.0/node_modules/
/websocket/build'
  CXX(target) Release/obj.target/bufferutil/src/bufferutil.o
  SOLINK_MODULE(target) Release/obj.target/bufferutil.node
  COPY Release/bufferutil.node
  CXX(target) Release/obj.target/validation/src/validation.o
  SOLINK_MODULE(target) Release/obj.target/validation.node
  COPY Release/validation.node
make: Leaving directory `/home/ncslab/다운로드/Mobius-Mobius-2.0.0/node_modules/
/websocket/build'
npm [notice] created a lockfile as package-lock.json. You should commit this file.
npm [WARN] Mobius@2.4.2 No repository field.

added 157 packages in 11.747s
ncslab@ncslab-System-Product-Name:~/다운로드/Mobius-Mobius-2.0.0$
```

위와 같이 “npm install” 을 입력하고 엔터를 치면 아래와 같이 필요한 모듈들이 설치가 된다. 만약 권한으로 오류가 발생하면 “sudo npm install” 로 설치한다.

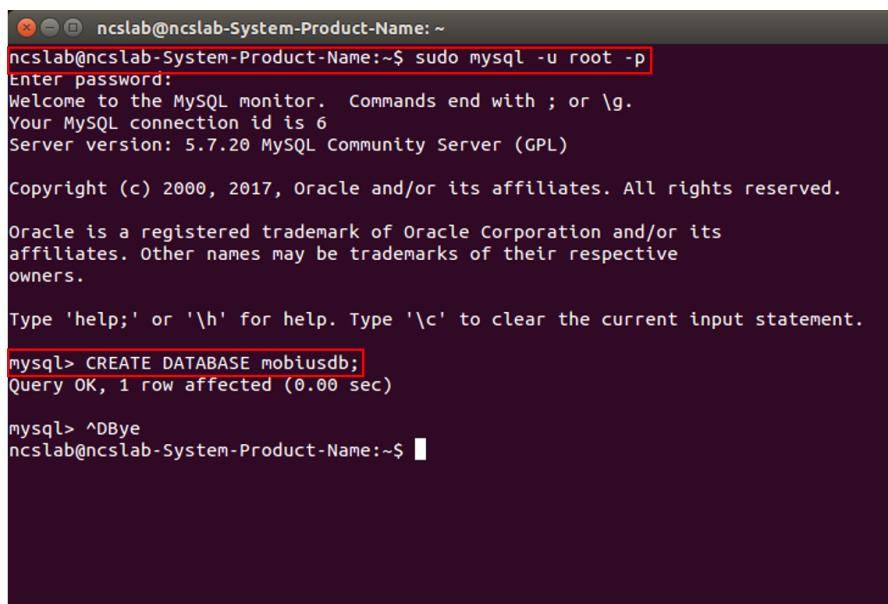
```
async-limiter@1.0.0
├── ultron@1.1.1
└── xtend@4.0.1
mysql@2.15.0
├── bignumber.js@4.0.4
└── sqlstring@2.3.0
shortid@2.2.8
url@0.11.0
├── punycode@1.3.2
└── querystring@0.2.0
util@0.10.3
├── inherits@2.0.1
└── websocket@1.0.25
  └── nan@2.8.0
    ├── typedarray-to-buffer@3.1.2
    │   └── is-typedarray@1.0.0
    └── yaeti@0.0.6
  └── xml2js@0.4.19
    ├── sax@1.2.4
    └── xmlbuilder@9.0.4
npm [WARN] Mobius@2.0.0 No repository field.
```

이렇게 설치가 되면 다음 그림과 같이 node_modules 폴더가 생긴 것을 볼 수 있다.

이제 모듈 설치가 끝났으니 다음과 같이 “node mobius.js”를 실행하면 Mobius 서버 플랫폼 설치가 완료된다.

2.3.2. *mobiusdb* 데이터베이스 생성(Linux)

다운로드 받은 Mobius 소스코드 디렉토리안에 있는 Mobius 디렉토리에 mobiusdb.sql 파일이 있다 이 파일을 통해 데이터베이스를 구성한다.



```
ncslab@ncslab-System-Product-Name: ~
ncslab@ncslab-System-Product-Name:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.20 MySQL Community Server (GPL)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

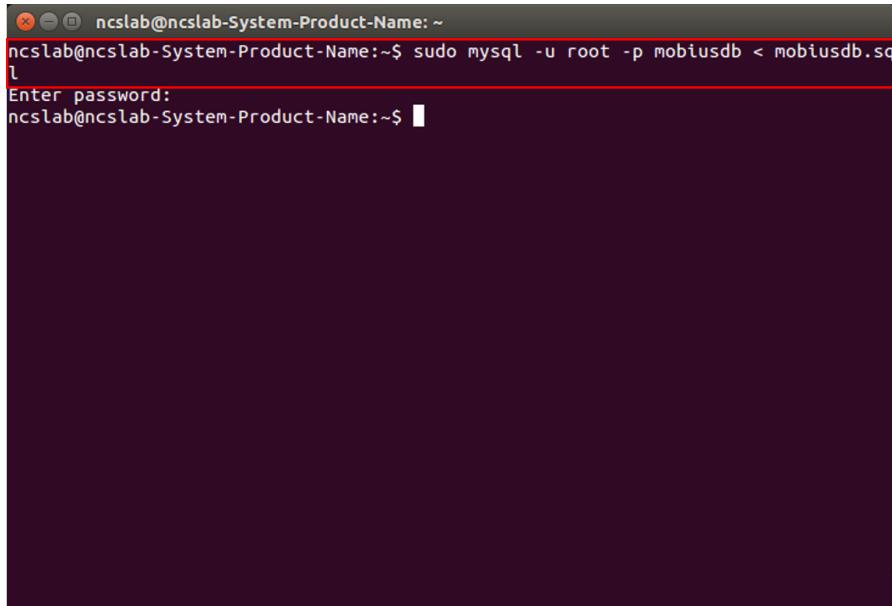
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE mobiusdb;
Query OK, 1 row affected (0.00 sec)

mysql> ^D
Bye
ncslab@ncslab-System-Product-Name:~$
```

먼저 데이터베이스를 생성하기에 앞서 “sudo mysql –u root –p” 명령어를 입력하여 root 으로 로그인을 한다. 입력하면 패스워드를 입력하라는 메시지를 볼 수 있는데 Mysql을 설치할 때 생성했던 패스워드를 입력한다. 접속이 되면 “CREATE DATABASE mobiusdb;”를 입력하여 데이터베이스를 생성한다. 생성이 되었는지 “SHOW DATABASES;”를 입력하면 확인할 수 있다.



다음으로 생성한 Mobius DB 테이블 생성을 위해 mobiusdb.sql 파일이 위치한 \$mobius-2.0/mobius 디렉토리로 이동한 후에 “sudo mysql –u root –p mobiusdb < mobiusdb.sql” 명령어를 입력하면 mobiusdb에 테이블이 구성된다.

```

ncslab@ncslab-System-Product-Name:~/mobius
ncslab@ncslab-System-Product-Name:~/mobius$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 5.7.20 MySQL Community Server (GPL)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE mobiusdb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>

```

Tables_in_mobiusdb
acp
ae
cb
cin
cnt
csr
grp
lcp
lookup
mgo
mms
nod
req
smd
sri
sub
ts
tsi

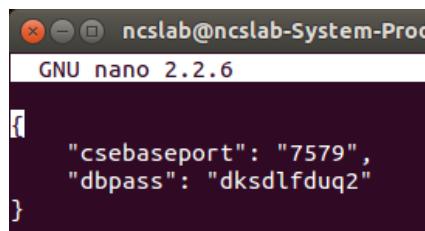
DB에 정상적으로 테이블이 생성되었는지 확인을 위해 mysql에 로그인하여 “USE mobiusdb;” 를 입력하여 DB를 선택한다. 선택된 mobiusdb안에 “SHOW TABLES;”를 통해 테이블이 구성되어있는 것을 확인 수 있다.

3. Mobius 서버 플랫폼 구동

Mobius 서버 설치까지 정상적으로 설치된 후 Mobius를 구동하기 전에 환경 설정을 해야 정상적으로 구동할 수 있다.

3.1. 환경설정

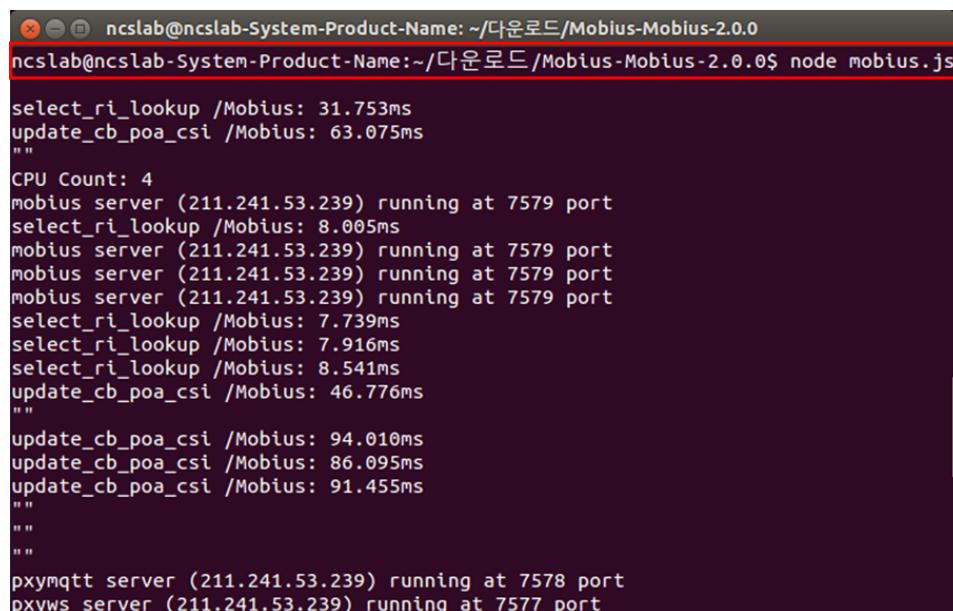
Mobius의 환경 설정은 conf.json 파일을 통해 설정을 변경할 수 있다.



```
{
  "csebaseport": "7579",
  "dbpass": "dksdlfduq2"
}
```

Mobius 구동 시 필요한 설정은 위와 같이 Mobius가 오픈하는 포트 번호와 환경구축 시 설치한 MySQL 데이터베이스 접근 암호를 설정해줘야 한다.

위의 정보를 구축하고자 하는 설정 정보로 업데이트를 한 후 아래와 같이 설치된 폴더에서 "node mobius.js" 명령어를 실행하여 Mobius 서버 플랫폼을 구동할 수 있다.



```
ncslab@ncslab-System-Product-Name:~/다운로드/Mobius-Mobius-2.0.0$ node mobius.js
select_ri_lookup /Mobius: 31.753ms
update_cb_poa_csi /Mobius: 63.075ms
"
CPU Count: 4
mobius server (211.241.53.239) running at 7579 port
select_ri_lookup /Mobius: 8.005ms
mobius server (211.241.53.239) running at 7579 port
mobius server (211.241.53.239) running at 7579 port
mobius server (211.241.53.239) running at 7579 port
select_ri_lookup /Mobius: 7.739ms
select_ri_lookup /Mobius: 7.916ms
select_ri_lookup /Mobius: 8.541ms
update_cb_poa_csi /Mobius: 46.776ms
"
update_cb_poa_csi /Mobius: 94.010ms
update_cb_poa_csi /Mobius: 86.095ms
update_cb_poa_csi /Mobius: 91.455ms
"
"
"
pxymqtt server (211.241.53.239) running at 7578 port
pxyws server (211.241.53.239) running at 7577 port
```

추가적으로 설정을 변경하고 싶다면 직접 소스를 변경하여야 한다. 추가적인 설정변경은 mobius.js 파일을 통해 이루어 진다. mobius.js 파일에 보면 다음과 같다.

```

var fs = require('fs');

var data = fs.readFileSync('conf.json', 'utf-8');
var conf = JSON.parse(data);

global.defaultmimetype = 'short';
global.defaultbodytype = 'json';

// my CSE information
global.usecsetype = 'in'; // select 'in' or 'mn' or 'asn'
global.usecsebase = 'Mobius';
global.usecseid = '/Mobius';
global.usecsebaseport = conf.csebaseport;

global.usedbhost = 'localhost';
global.usedbpass = conf.dbpass;

global.usepxywsport = '7577';
global.usepxymqtport = '7578';

global.usetsagentport = '7582';

global.usemqttbroker = 'localhost'; // mqttbroker for mobius
global.useselector = '10.10.202.114';
global.usescure = 'disable';
if(usescure === 'enable') {
    global.usemqttport = '8883';
}
else {
    usemqttport = '1883';
}

```

mobius.js 서버의 정보 설정

- usecsetype: 서버 타입, Mobius 는 IN-CSE 로써 'in'으로 입력되어야 한다.
- usecsebase: Mobius 에 대한 CSEBase 이름을 나타낸다. 디폴트로 mobius-
yt 로 되어 있다.
- usecseid: Mobius 에 대한 cseid 를 입력한다. cseid 는 '/'로 시작하여야
하며, 디폴트로 csebase 이름 앞에 '/'를 붙여서 사용한다.
- usedbhost: DB 서버 주소, 디폴트로 localhost 를 사용한다.
- usemqttbroker: mqtt broker 호스트 주소, 디폴트로 localhost 를 사용한다.

3.2. 테스트

3.2.1. 리소스뷰어(oneM2M Browser) 활용

Mobius와 nCube를 통해 만들어진 디바이스가 제대로 동작하는지 테스트하기 위해 웹으로 제공하는 리소스뷰어를 사용할 수 있다. 리소스뷰어는 윈도우 파일 탐색기와 비슷하게 트리 구조의 리소스를 보여주고 리소스 생성, 삭제의 간단한 동작을 수행할 수 있다.

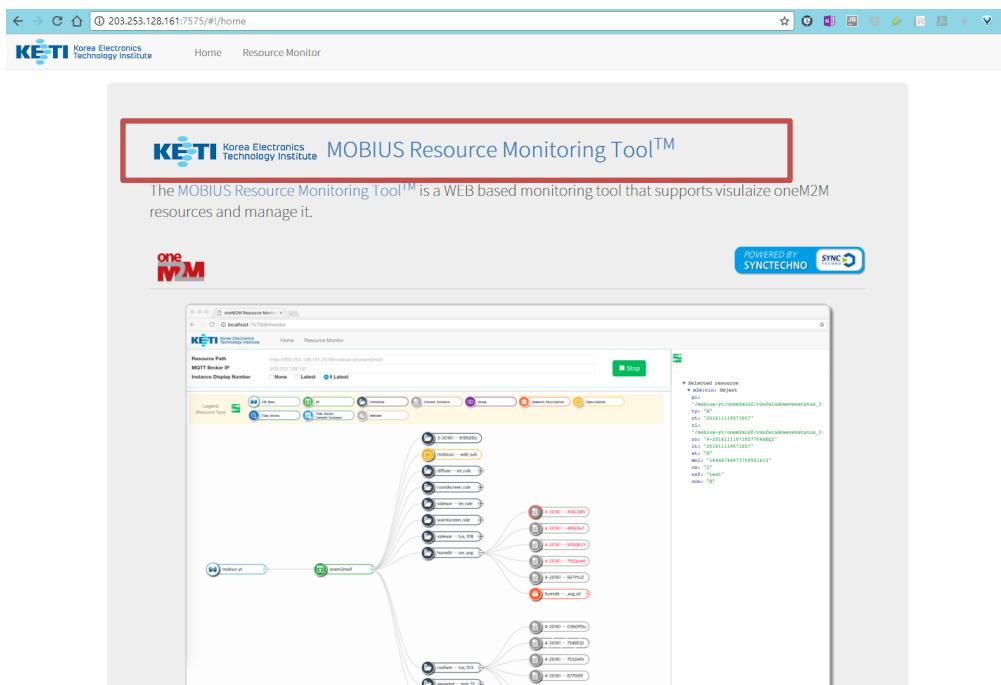
웹버전의 리소스뷰어는 아래의 url로 제공하고 있다.

<http://203.253.128.161:7575>

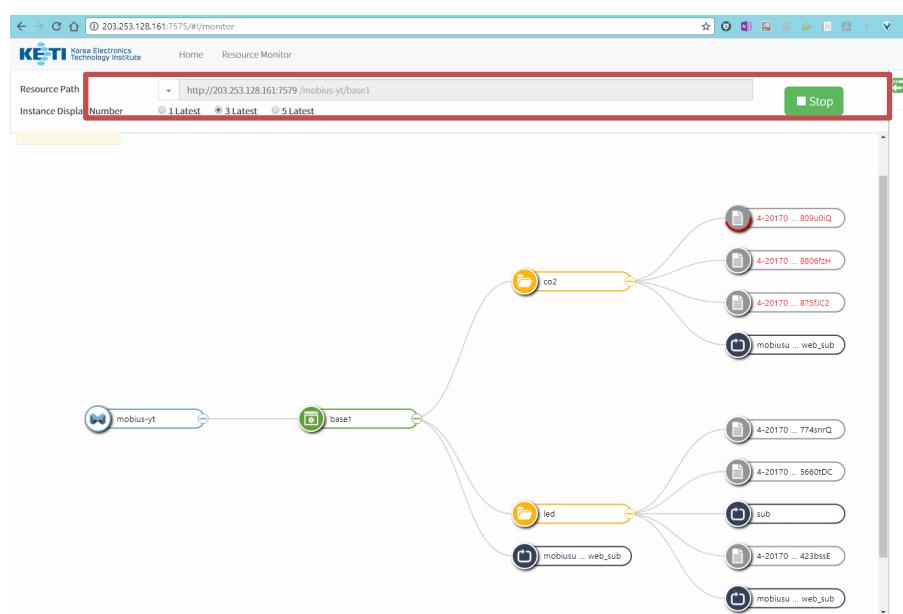
웹버전의 리소스뷰어 Mobius의 리소스만을 볼 수 있는 도구이고 다른 oneM2M 플랫폼의 구조를 볼 수 있는 버전은 OCEAN의 GitHub에 윈도우 응용 프로그램 버전의

oneM2M Browser로 제공하고 있다.

위의 주소로 접속하면 아래와 같이 리소스뷰어 홈페이지가 출력된다. 메뉴의 Sresource Monitor를 선택하거나 표시된 부분을 선택하면 리소스뷰어로 전환된다.



아래와 같이 리소스뷰어가 보이면 표시된 부분에 Mobius에 생성된 리소스의 url을 입력하고 Strat 버튼을 누르면 해당 url의 리소스 구조를 볼 수 있다. url 부분의 처음은 콤보 박스로 되어 있고 이전 Mobius와 Mobius release 2를 선택할 수 있다. 이후 부분은 직접 입력해야 하고 여기에 생성한 AE의 name을 입력하면 된다.



3.2.2. Postman 활용

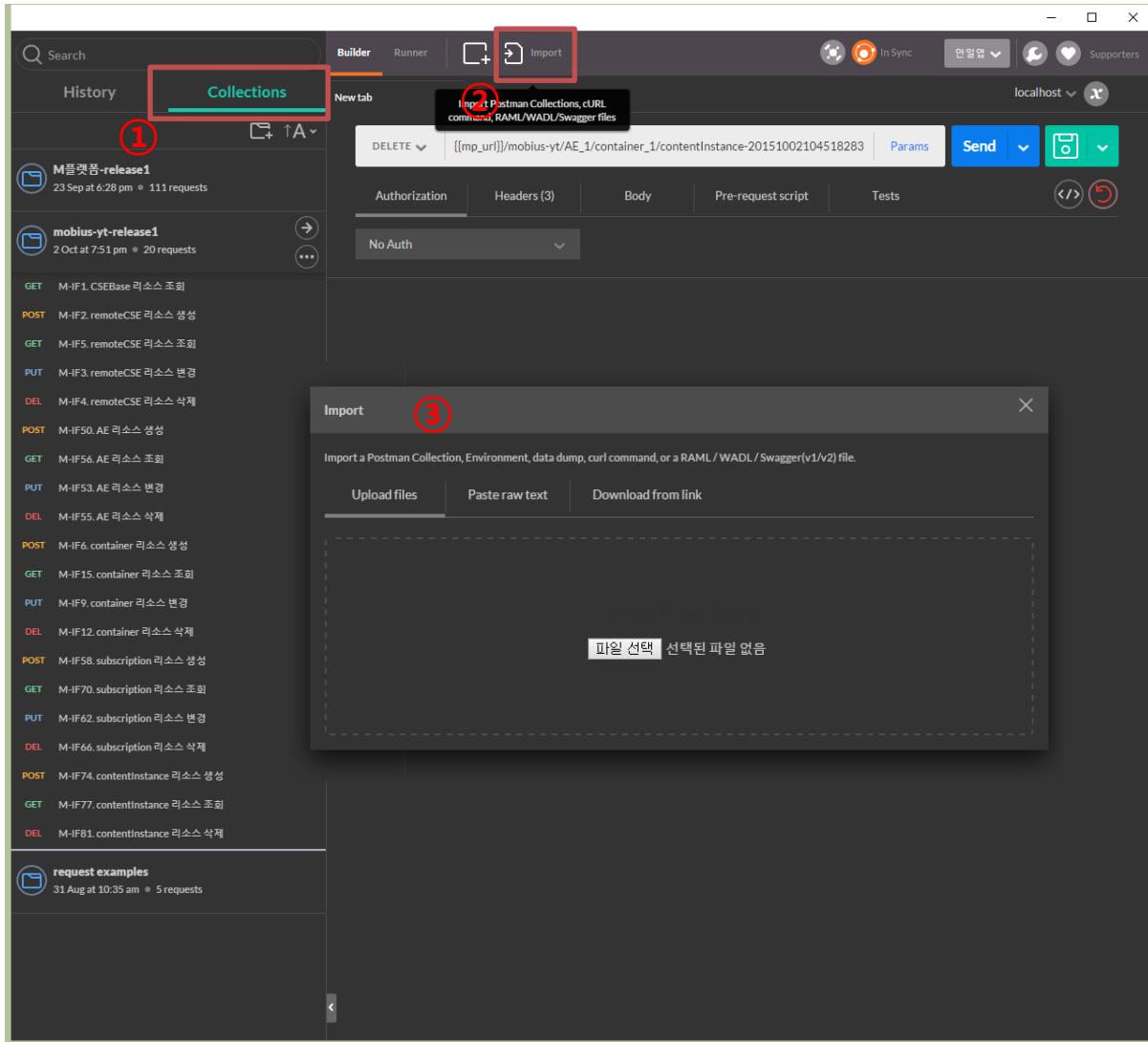
크롬 확장 프로그램인 Postman을 활용하여 직접 구축한 Mobius를 동작을 테스트 할 수 있다.

아래와 같이 크롬 브라우저에서 설정 메뉴를 선택하고 설정화면의 확장 프로그램 탭에서 “더 많은 확장 프로그램 다운로드”를 선택한다. 검색란에 “postman rest client”를 입력하고 Postman을 찾아 설치할 수 있다.



Postman을 설치하면 아래와 같이 실행이 된다. Mobius 테스트를 위해 OCEAN에서 미리

만들어 둔 스크립트를 다운받아 Import 한다. Import 하는 방법은 그림과 같이 Collections 탭을 선택하고 바로 오른쪽 위에 Import 메뉴를 선택한다. Import를 위한 창이 열리면 Upload files 탭의 파일 선택을 눌러 OCEAN에서 다운로드 받은 스크립트 파일을 선택하고 import 한다.



Import가 성공하면 아래 그림처럼 왼쪽에 Mobius-yt-release1 이라는 콜렉션 탭이 생성이 될 것이다. 이 테스트 스크립트는 현재 지원되는 리소스의 CRUD를 할 수 있도록 정리를 해 둔 것이고 xml 데이터 포맷으로 테스트 할 수 있도록 설정되어 있다.

The screenshot shows a list of API requests for the 'mobius-xt-release1' instance. The requests are categorized by method and endpoint:

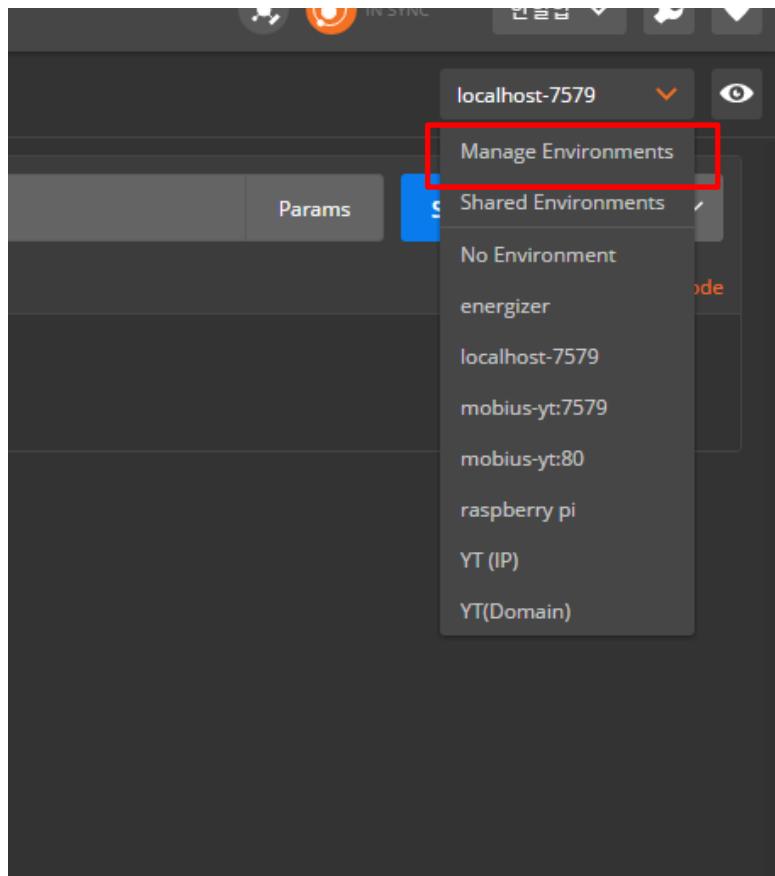
- GET**: M-IF1.CSEBase 리소스 조회, M-IF5.remoteCSE 리소스 조회, M-IF56.AE 리소스 조회, M-IF15.container 리소스 조회.
- POST**: M-IF2.remoteCSE 리소스 생성, M-IF50.AE 리소스 생성, M-IF6.container 리소스 생성, M-IF70.subscription 리소스 조회.
- PUT**: M-IF3.remoteCSE 리소스 변경, M-IF9.container 리소스 변경.
- DEL**: M-IF4.remoteCSE 리소스 삭제, M-IF53.AE 리소스 삭제, M-IF55.AE 리소스 삭제, M-IF12.container 리소스 삭제, M-IF62.subscription 리소스 변경, M-IF66.subscription 리소스 삭제.

테스트를 위해 직접 구축한 서버 정보를 입력해야 하는데 아래 그림과 같이 주소창에서 {{mp_url}} 값을 설정해 주어야 한다. {{mp_url}} 대신에 http://(직접구축한서버주소):포트번호를 입력하여 사용할 수 있다.

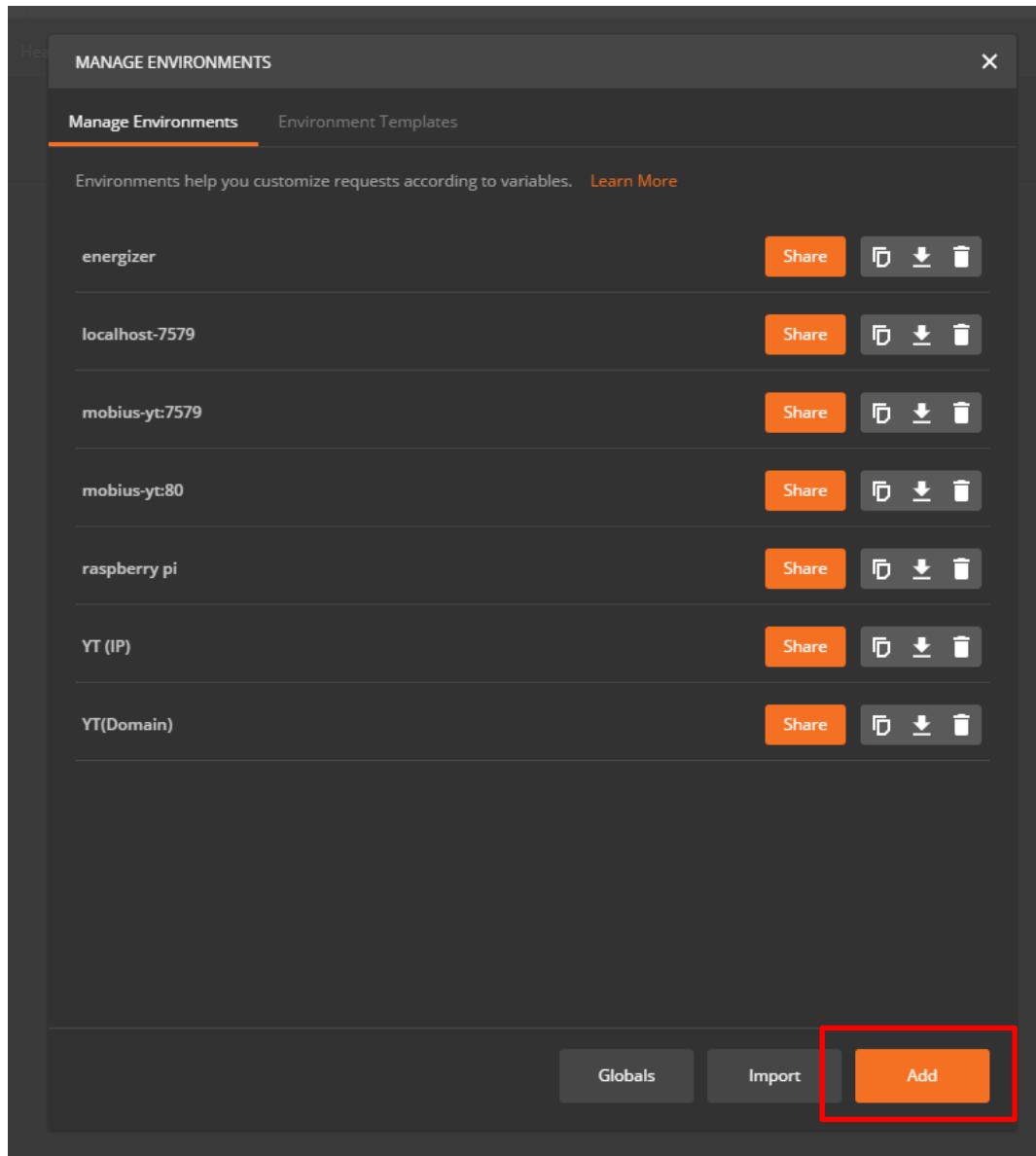
The screenshot shows the Postman interface for the 'M-IF1.CSEBase 리소스 조회' request. The configuration includes:

- Method**: GET
- URL**: http://203.253.174.104:7579/mobius-xt
- Authorization**: No Auth
- Headers**: (4)
- Body**: (empty)
- Pre-request script**: (empty)
- Tests**: (empty)

그러면, {{mp_url}} 설정하는 방법은 아래와 같다.



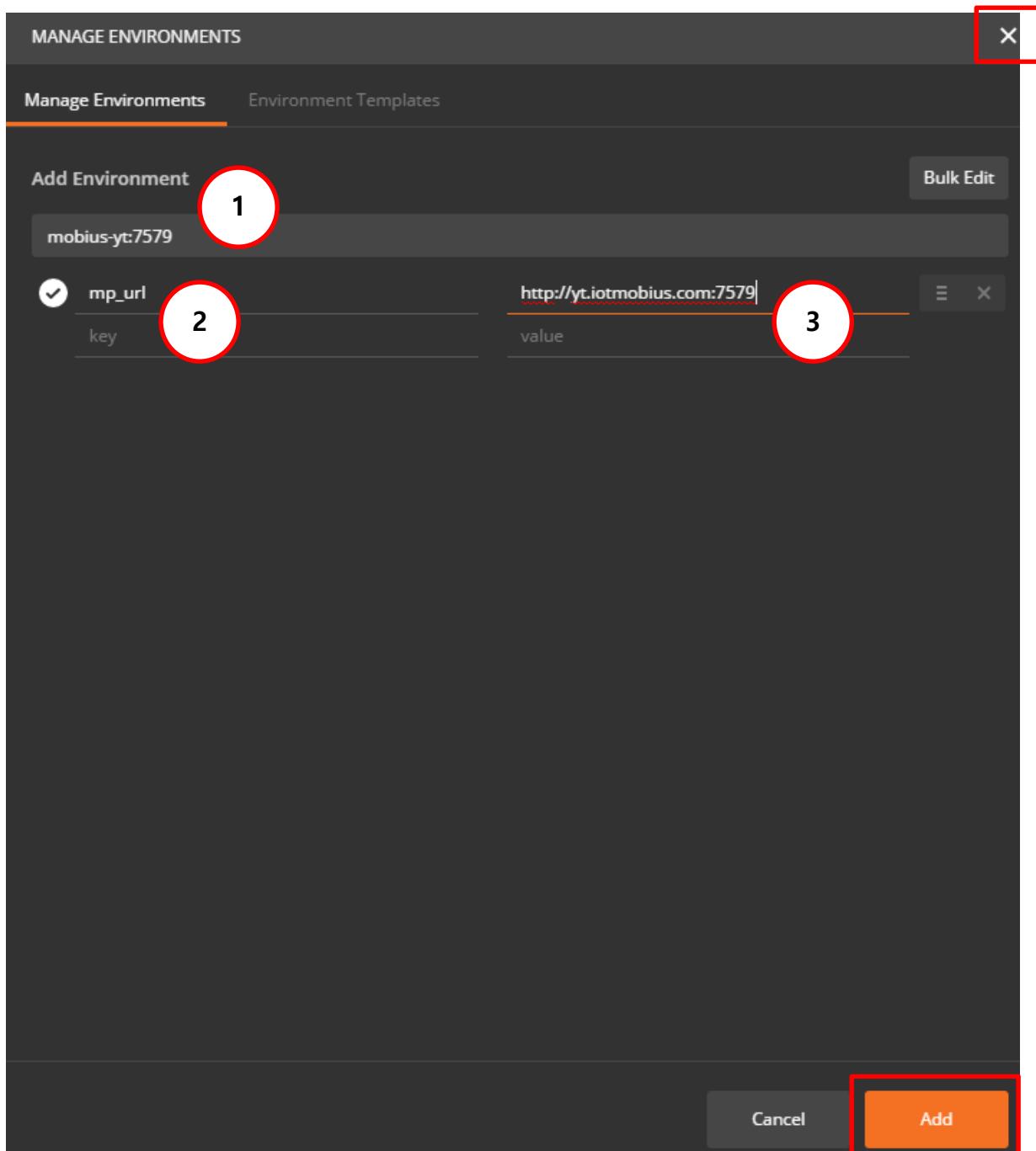
포스트맨 창에서 오른쪽 위에 보면 설정할 수 있는 리스트박스가 있고 이를 선택하면 그림과 같이 리스트 메뉴가 나온다. 이 때 Manage Environments 를 선택한다.



설정 창이 나오면 Add 버튼을 눌러 아래와 같이 {{mp_url}} 값을 입력한다.

- ① 설정에 대한 이름을 입력한다. (mobius-xt:7579)
- ② mp_url 이라고 입력한다.
- ③ 접속하고자 하는 서버 주소를 입력한다. (<http://203.253.128.161:7579>)

설정이 완료되면 Add 버튼을 눌러 추가하고 창을 닫는다.



다시 포스트맨 창으로 돌아와서 리스트 박스에서 방금 추가한 설정을 선택하면

이후 매번 주소를 입력할 필요없이 배포한 스크립트를 사용할 수 있다.