



**Software Version: 2.5.12**

---

## Installation Guide v2.5.12

Document Release Date: Jan 2023

Software Release Date: Jan 2023



## Revision History

버전	변경일자	제·개정 내역
1.0.0		초안
1.1.0	2018-02-11	- SW 버전 업데이트에 따른 내용 추가
1.2.0	2022-05-09	- Mobius 구동 환경 변화에 따른 필수 SW 설치 내용 수정
2.5.12	2022-01-19	SW 버전 업데이트에 따른 내용 추가 및 MQTT 설정 내용 추가

wonseok@keti.re.kr / 정원석  
hyeonseo0128@keti.re.kr / 손현서  
eunbeelee@keti.re.kr / 이은비  
monto8790@gmail.com / 박윤기  
sm.jeong@keti.re.kr / 정승명  
iyahn@keti.re.kr / 안일엽

## 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 .....	6
1.1.	Mobius 란?.....	6
1.2.	Mobius 플랫폼 .....	7
1.2.1.	<i>Mobius</i> 서버 플랫폼 개요.....	7
1.2.2.	<i>Mobius</i> 서버 플랫폼 주요 기능.....	8
1.2.3.	<i>Mobius</i> 서버 플랫폼 구성.....	8
1.2.4.	<i>Mobius</i> 서버 플랫폼 S/W 아키텍처.....	9
1.2.5.	<i>Mobius</i> 서버 디렉토리 구조.....	10
2.	Mobius 서버 플랫폼 설치 .....	15
2.1.	설치 개요 .....	15
2.2.	Mobius 구동환경 구축 (Linux) .....	15
2.2.1.	<i>MySQL</i> 설치.....	15
2.2.2.	<i>MQTT</i> 서버 설치.....	19
2.2.3.	<i>Node JS</i> 설치.....	24
2.3.	Mobius 설치 .....	27
2.3.1.	다운로드 및 모듈 설치 .....	27
2.3.2.	<i>mobiusdb</i> 데이터베이스 생성(Linux).....	30
3.	Mobius 서버 플랫폼 구동 .....	32
3.1.	환경설정 .....	32
3.2.	테스트 .....	35
3.2.1.	리소스뷰어( <i>oneM2M Browser</i> ) 활용 .....	35
3.2.2.	<i>Postman</i> 활용 .....	36

# 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.

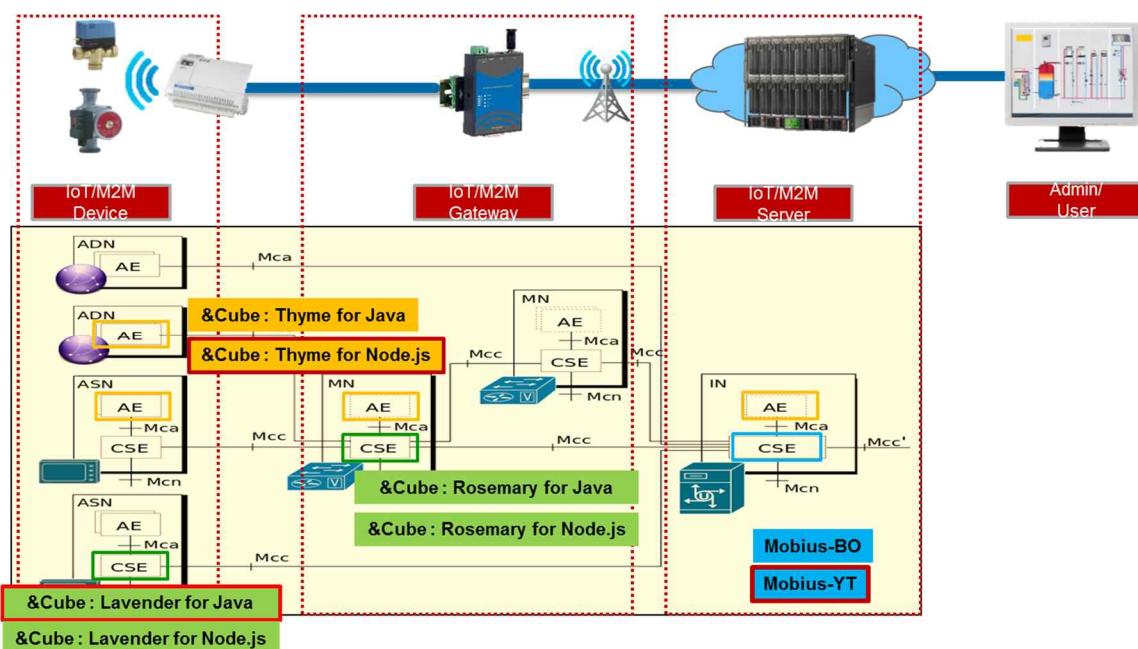


Figure 1 oneM2M 기반 Mobius 플랫폼 개념도

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

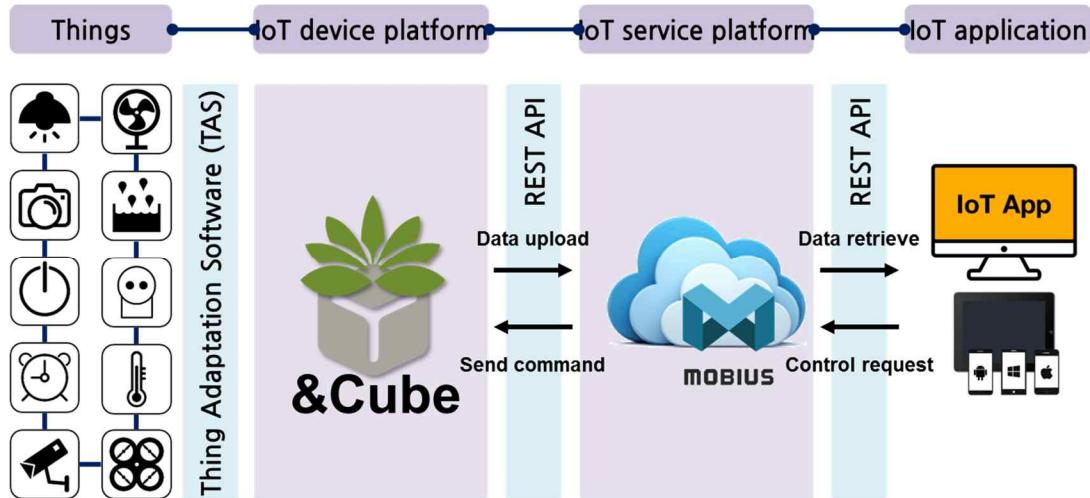


Figure 2 Mobius 와 IoT 디바이스 연동

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.

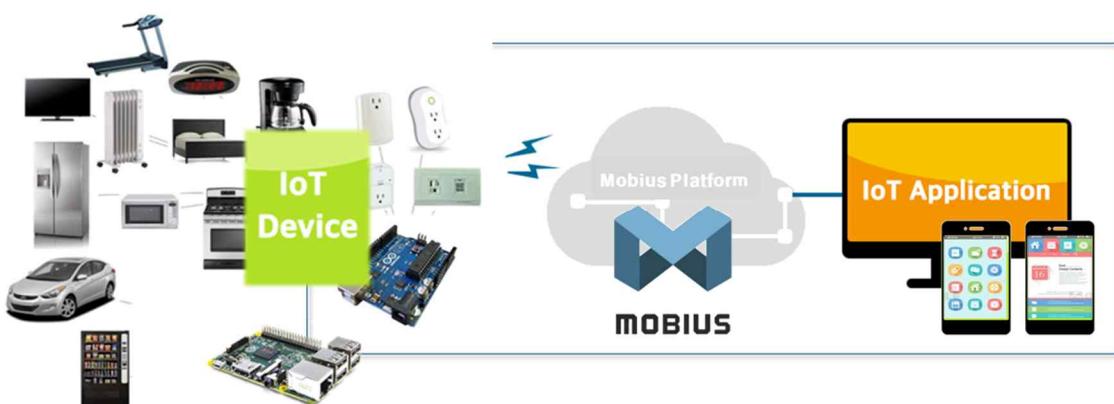


Figure 3 Mobius Platform 개요

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

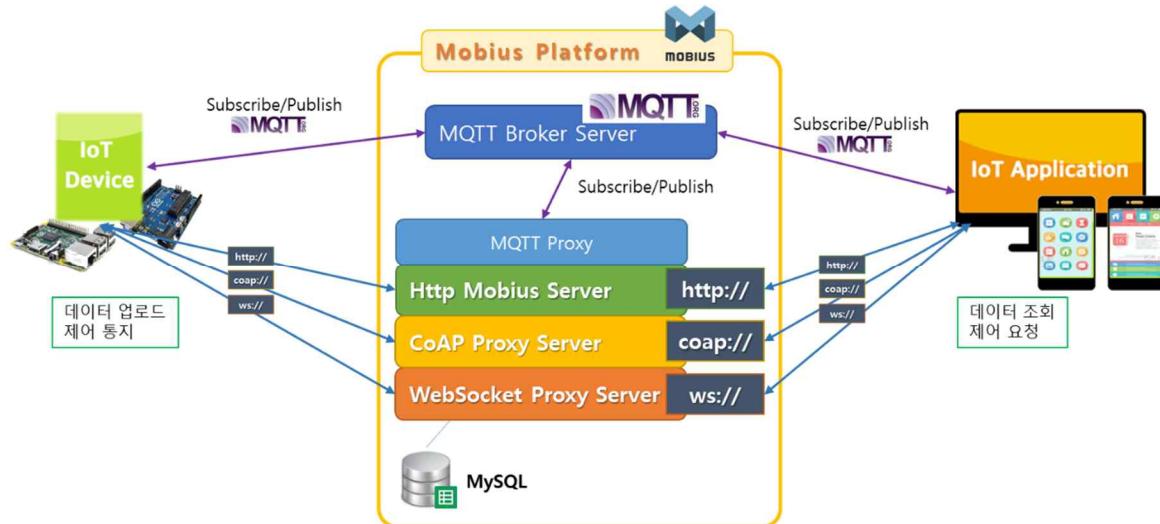


Figure 5 Mobius 서버 플랫폼 시스템 구성

#### 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.

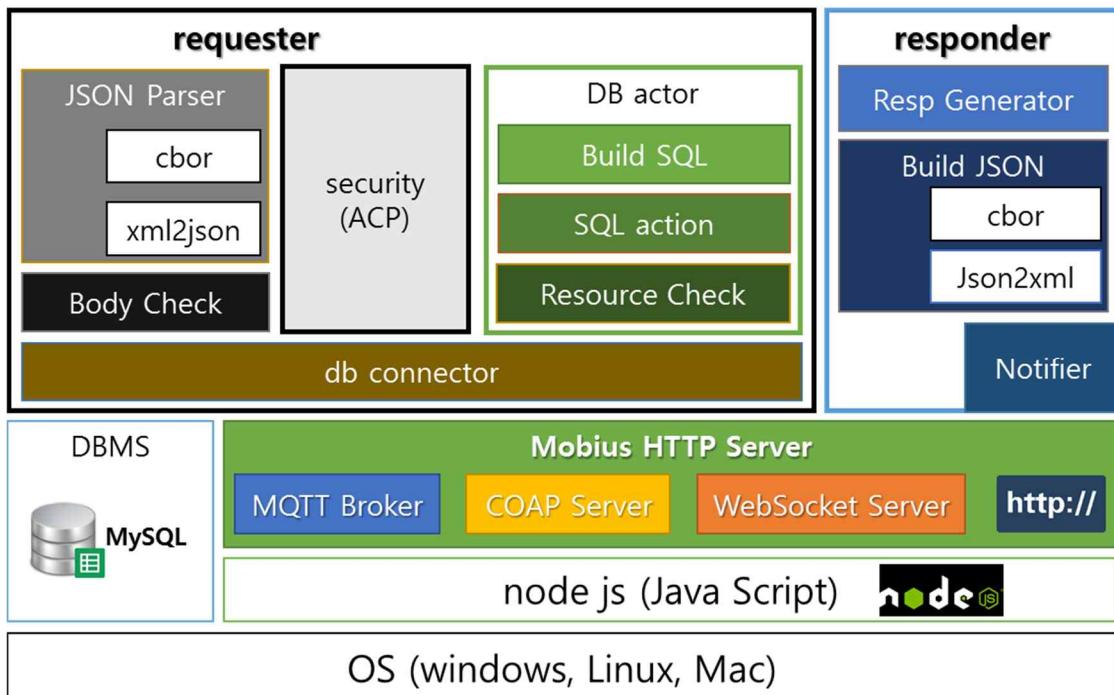


Figure 6 Mobius S/W 아키텍처

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 두 부분으로 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.

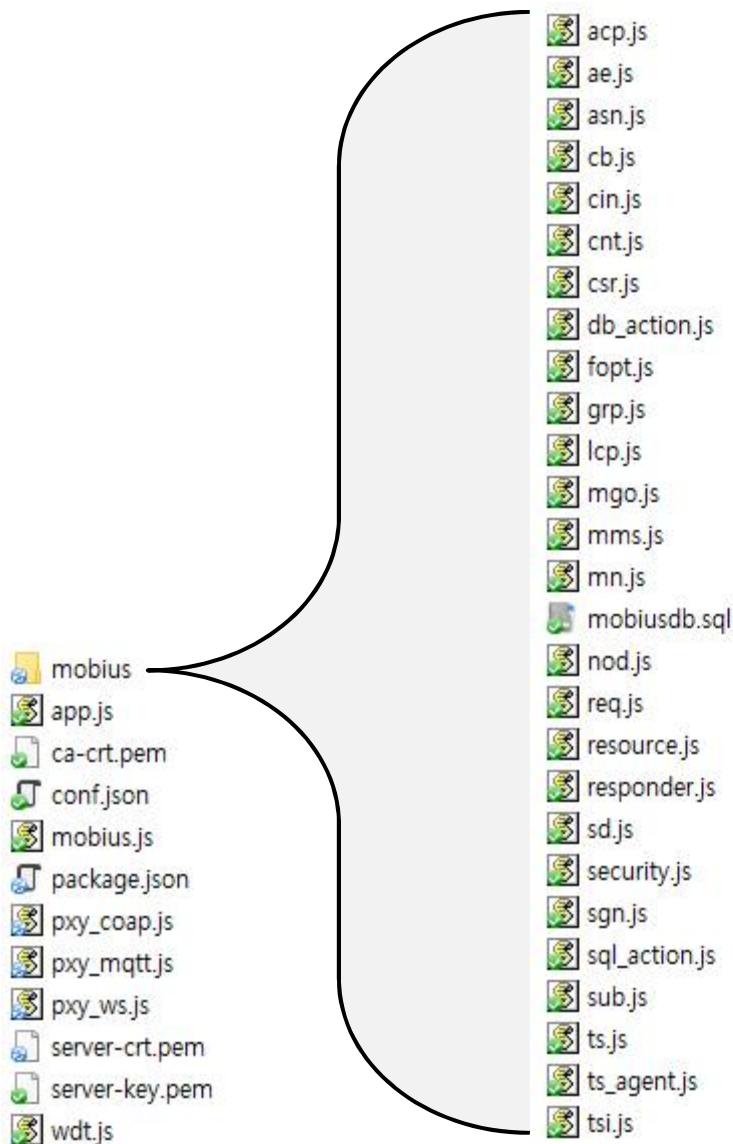
[http://hostname:port/IN-CSE\\_name/AE\\_name/container\\_name/contentInstance\\_name](http://hostname:port/IN-CSE_name/AE_name/container_name/contentInstance_name)

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 값을 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 response를 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 response를 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 response를 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 20.04 버전으로 진행하였다.

<http://www.mysql.com>

## ④ MySQL Community Downloads

◀ MySQL Community Server



```
son@son-linux:~$ wget https://dev.mysql.com/get/mysql-apt-config_0.8.24-1_all.deb
--2023-01-18 11:52:24-- https://dev.mysql.com/get/mysql-apt-config_0.8.24-1_all.deb
Resolving dev.mysql.com (dev.mysql.com)... 104.76.87.97, 2600:1410:2000:1ba::2e31, 2600:1410:2000:194::2e31
Connecting to dev.mysql.com (dev.mysql.com)|104.76.87.97|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://repo.mysql.com/mysql-apt-config_0.8.24-1_all.deb [following]
--2023-01-18 11:52:24-- https://repo.mysql.com/mysql-apt-config_0.8.24-1_all.deb
Resolving repo.mysql.com (repo.mysql.com)... 104.118.100.221
Connecting to repo.mysql.com (repo.mysql.com)|104.118.100.221|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18048 (18K) [application/x-debian-package]
Saving to: 'mysql-apt-config_0.8.24-1_all.deb'

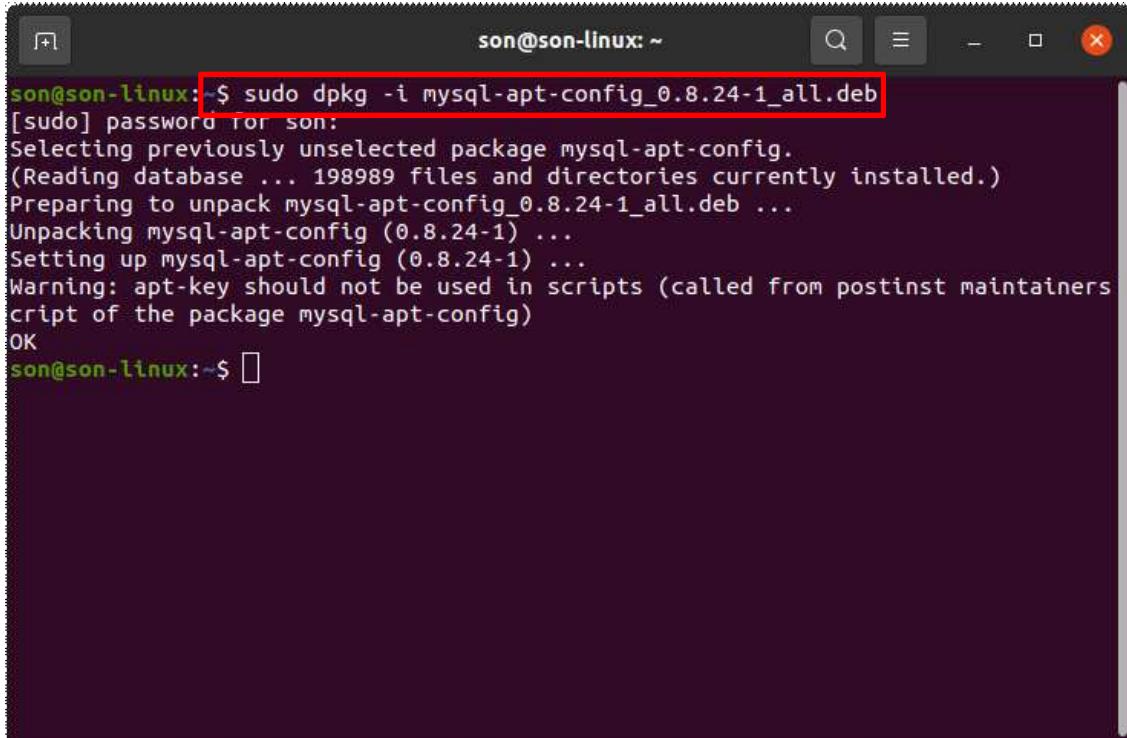
mysql-apt-config_0. 100%[=====] 17.62K --- KB/s in 0s

2023-01-18 11:52:25 (54.3 MB/s) - 'mysql-apt-config_0.8.24-1_all.deb' saved [18048/18048]

son@son-linux:~$
```

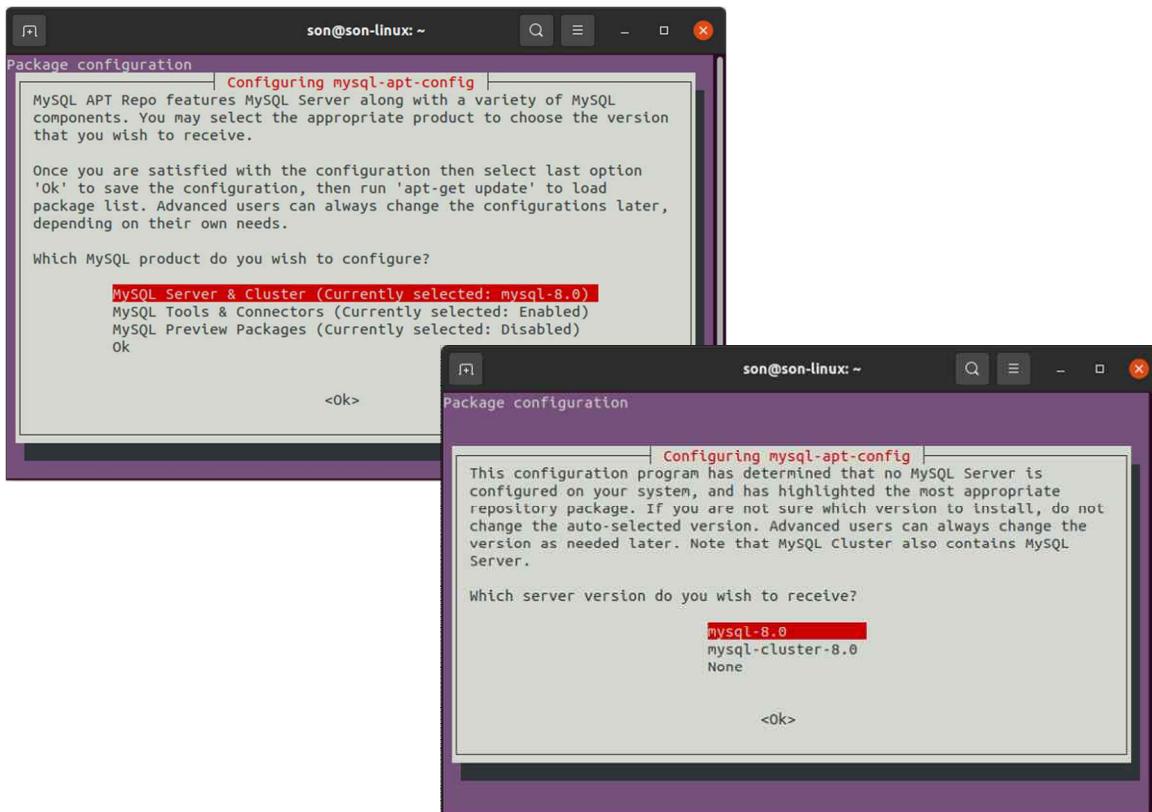
우분투 버전 18.04는 서버로부터 MySQL 패키지를 직접 다운받아 설치리스트에 추가해야 한다. “wget http://dev.mysql.com/get/mysql-apt-config\_0.8.24-1\_all.deb” 다음과 같은 명령어를 통해 다운로드한다.

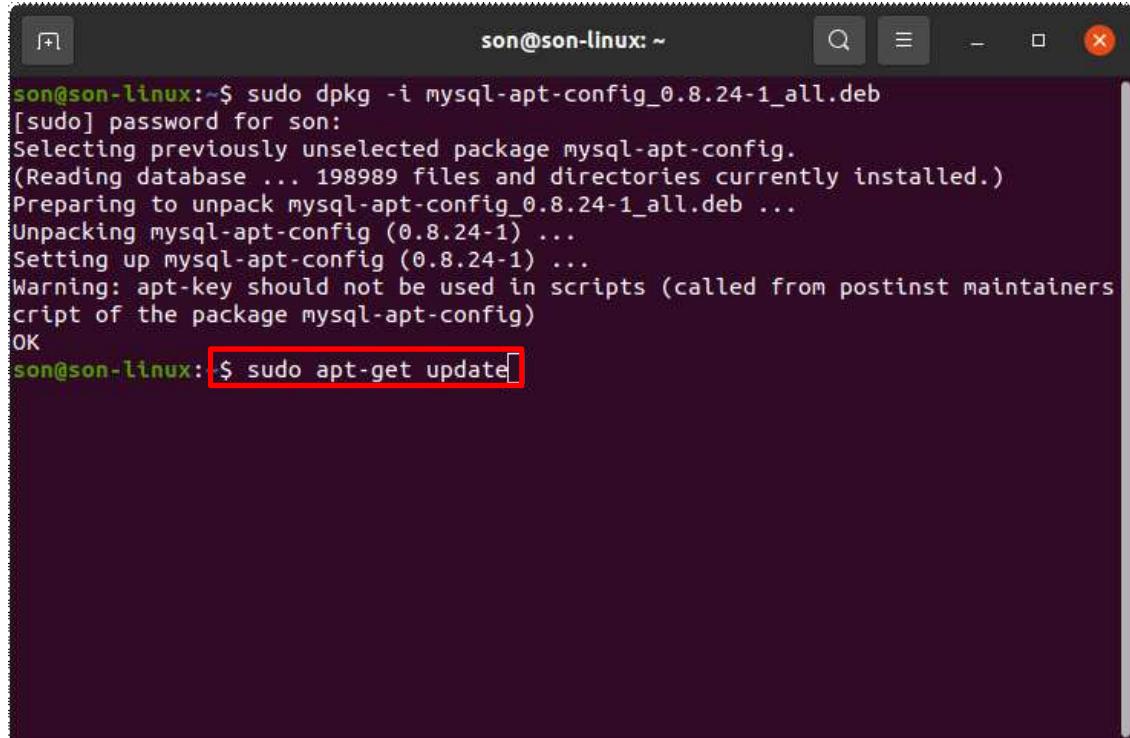
우분투 버전 20.04는 패키지 업데이트 및 업그레이드만으로도 MySQL 8.0 설치가 가능하다.



```
son@son-linux:~$ sudo dpkg -i mysql-apt-config_0.8.24-1_all.deb
[sudo] password for son:
Selecting previously unselected package mysql-apt-config.
(Reading database ... 198989 files and directories currently installed.)
Preparing to unpack mysql-apt-config_0.8.24-1_all.deb ...
Unpacking mysql-apt-config (0.8.24-1) ...
Setting up mysql-apt-config (0.8.24-1) ...
Warning: apt-key should not be used in scripts (called from postinst maintainer's
script of the package mysql-apt-config)
OK
son@son-linux:~$
```

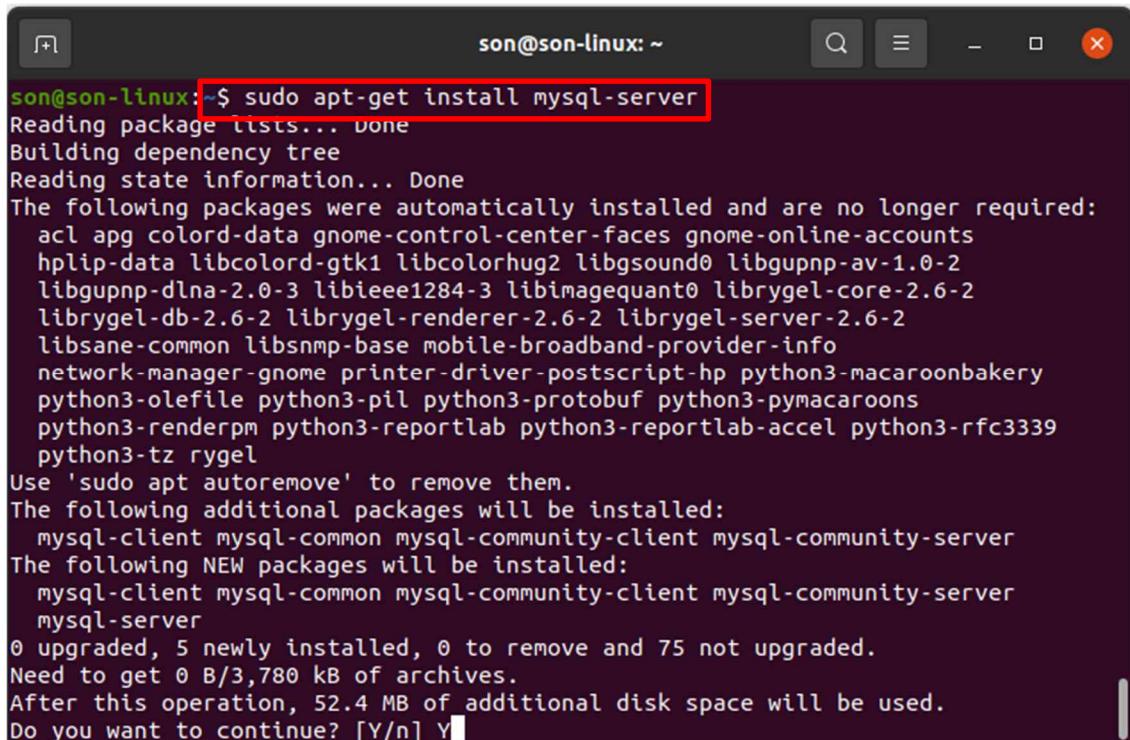
다운받은 deb 파일을 “dpkg” 명령어를 사용하여 설치합니다. 명령어를 입력하면 설치하려는 MySQL 을 선택하는 화면이 나옵니다. MySQL Server(mysql-8.0)를 선택하고 ENTE 를 눌러 8.0 버전 선택한 후, Apply 를 선택하여 ENTER 을 누른다.





```
son@son-linux:~$ sudo dpkg -i mysql-apt-config_0.8.24-1_all.deb
[sudo] password for son:
Selecting previously unselected package mysql-apt-config.
(Reading database ... 198989 files and directories currently installed.)
Preparing to unpack mysql-apt-config_0.8.24-1_all.deb ...
Unpacking mysql-apt-config (0.8.24-1) ...
Setting up mysql-apt-config (0.8.24-1) ...
Warning: apt-key should not be used in scripts (called from postinst maintainers
script of the package mysql-apt-config)
OK
son@son-linux:~$ sudo apt-get update
```

"sudo apt-get update" 명령어를 통해 설치 패키지 리스트를 최신화한다.



```
son@son-linux:~$ sudo apt-get install mysql-server
Reading package lists... done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  acl apg colord-data gnome-control-center-faces gnome-online-accounts
  hplip-data libcolord-gtk1 libcolorhug2 libgsound0 libgupnp-av-1.0-2
  libgupnp-dlna-2.0-3 libieee1284-3 libimagequant0 librygel-core-2.6-2
  librygel-db-2.6-2 librygel-renderer-2.6-2 librygel-server-2.6-2
  libsane-common libsnmp-base mobile-broadband-provider-info
  network-manager-gnome printer-driver-postscript-hp python3-macaroonbakery
  python3-olefile python3-pil python3-protobuf python3-pymacaroons
  python3-renderpm python3-reportlab python3-reportlab-accel python3-rfc3339
  python3-tz rygel
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  mysql-client mysql-common mysql-community-client mysql-community-server
The following NEW packages will be installed:
  mysql-client mysql-common mysql-community-client mysql-community-server
  mysql-server
0 upgraded, 5 newly installed, 0 to remove and 75 not upgraded.
Need to get 0 B/3,780 kB of archives.
After this operation, 52.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

"sudo apt-get install mysql-server" 입력하여 설치를 진행한다.

```

son@son-linux:~$ sudo /usr/bin/mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.33 MySQL Community Server - GPL

mysql> SHOW VARIABLES LIKE "%Version%";
+-----+-----+
| Variable_name      | Value   |
+-----+-----+
| version            | 8.0.33  |
| version_comment    | MySQL Community Server - GPL |
| version_compile_machine | x86_64 |
| version_compile_os  | Linux   |
| version_compile_zlib | 1.2.13  |
+-----+-----+
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'keti123';
Query OK, 0 rows affected (0.01 sec)

mysql>

```

\*\*mysql 기본 설치 dir = /usr/bin/mysql

Mysql 설치가 모두 완료되었다면 이후 “sudo [mysql 설치 dir] -u root -p” 명령어를 통해 mysql 접속을 실행한다.

그리고 “SHOW VARIABLES LIKE “%Version%”;” 명령어를 통해 설치된 mysql의 버전을 확인한다.

mysql 버전에 맞춰 다음과 같은 Query 명령어를 통해 mysql 접속 비밀번호를 변경한다.

Version이 5.7.6 이후 → ALTER USER 'root'@'localhost' IDENTIFIED BY '설정 비밀번호';

Version이 5.7.6 이전 → SET PASSWORD FOR 'root'@'localhost' = PASSWORD('설정 비밀번호');

해당 비밀번호는 이후 Mobius와 mysql 연동 시 사용될 계정에 대한 비밀번호로

위의 예시에선 'localhost'라는 dbhost에 'keti123'이라는 pw를 설정한 예시이다.

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

<https://mosquitto.org>

Download 페이지에 설명되어 있는 설치 방법대로 설치한다.

The screenshot shows the Mosquitto project website. At the top, there are logos for mosquitto, ECLIPSE FOUNDATION, and cedalo. To the right are links for Home, Blog, Download, and Documentation. Below these, a "Download" button is prominent. Under "Source", there are two bullet points: "mosquitto-2.0.15.tar.gz (GPG signature)" and "Git source code repository (github.com)". A note below says "Older downloads are available at <https://mosquitto.org/files/>".

## Binary Installation

The binary packages listed below are supported by the Mosquitto project. In many cases Mosquitto is also available directly from official Linux/BSD distributions.

### Debian

- Mosquitto is now in Debian proper. There will be a short delay between a new release and it appearing in Debian as part of the normal Debian procedures.
- There are also Debian repositories provided by the mosquitto project, as described at <https://mosquitto.org/2013/01/mosquitto-debian-repository>

### Raspberry Pi

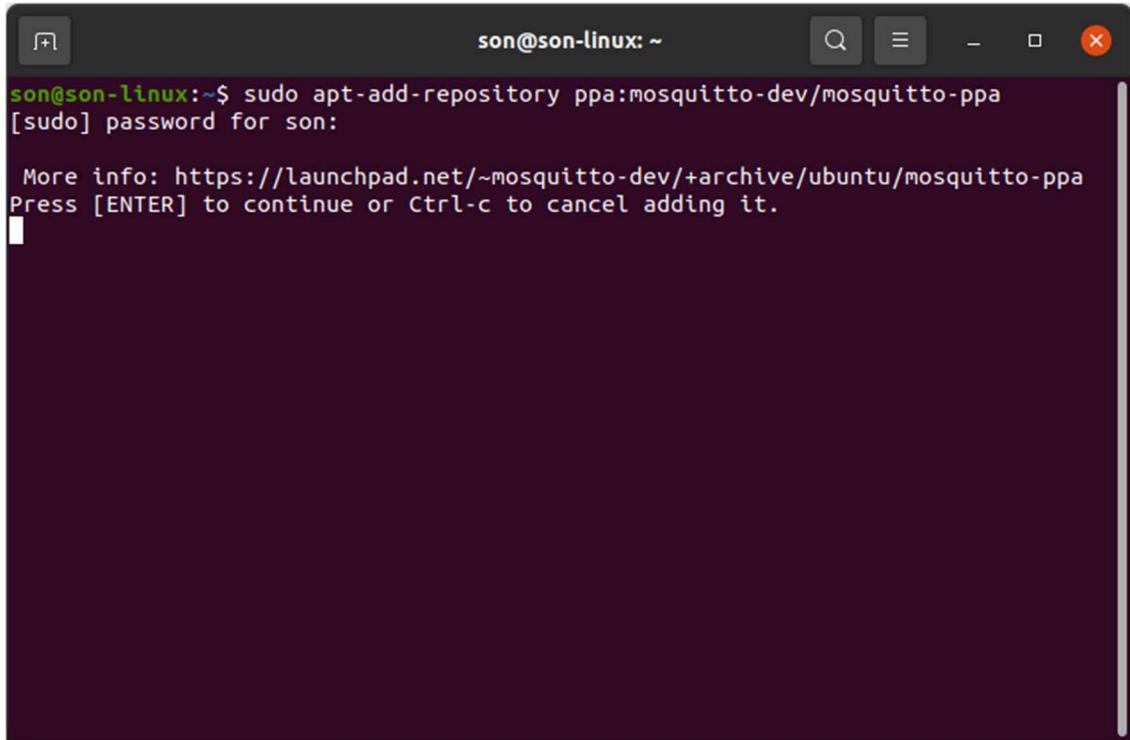
Mosquitto is available through the main repository.

There are also Debian repositories provided by the mosquitto project, as described at <https://mosquitto.org/2013/01/mosquitto-debian-repository/>

### Ubuntu

Mosquitto is available in the Ubuntu repositories so you can install as with any other package. If you are on an earlier version of Ubuntu or want a more recent version of mosquitto, add the [mosquitto-dev PPA](#) to your repositories list - see the link for details. mosquitto can then be installed from your package manager.

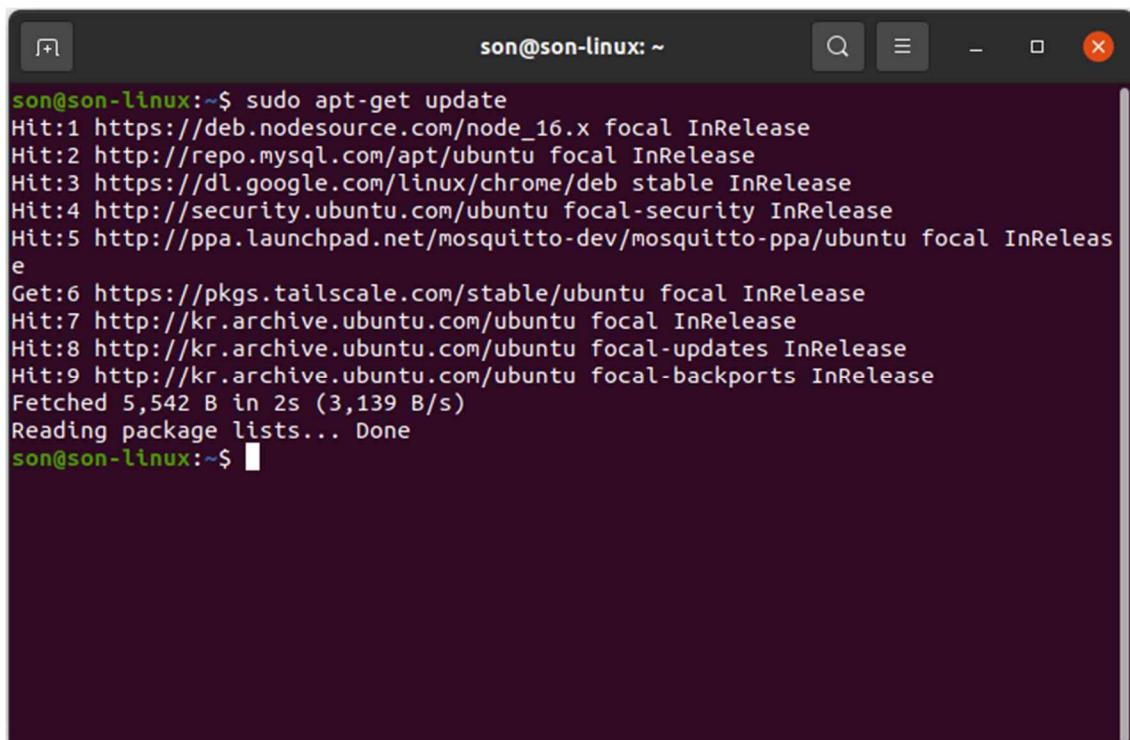
- `sudo apt-add-repository ppa:mosquitto-dev/mosquitto-ppa`
- `sudo apt-get update`



```
son@son-linux:~$ sudo apt-add-repository ppa:mosquitto-dev/mosquitto-ppa
[sudo] password for son:

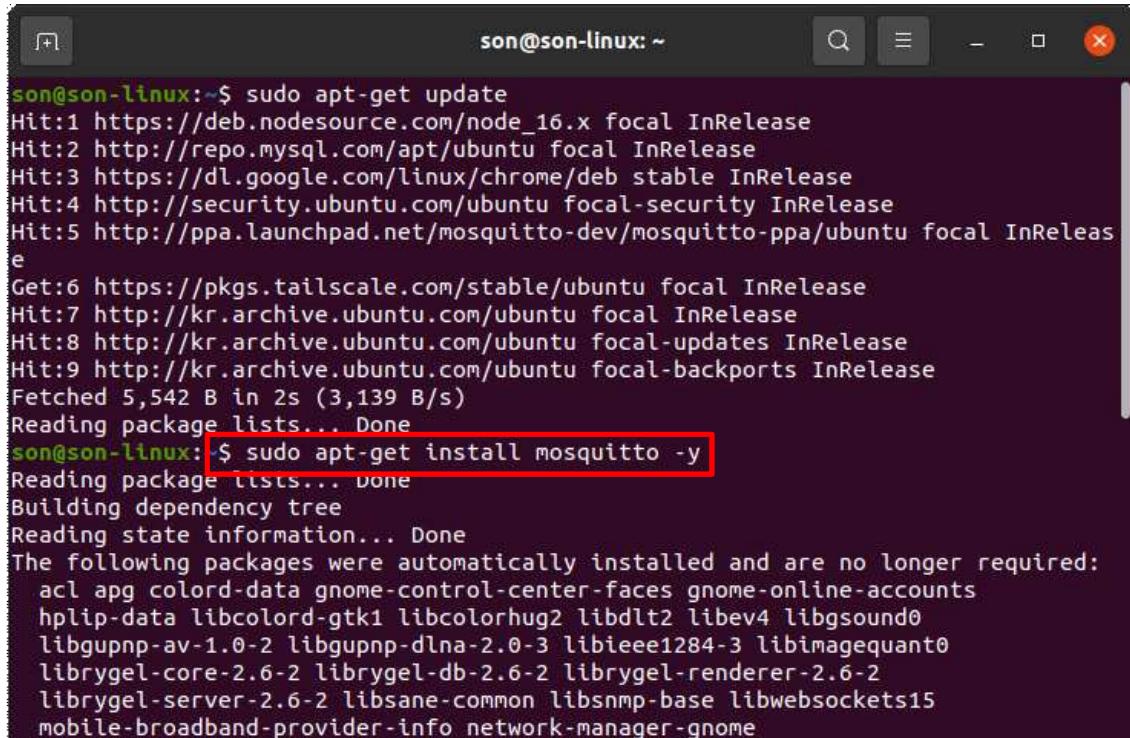
More info: https://launchpad.net/~mosquitto-dev/+archive/ubuntu/mosquitto-ppa
Press [ENTER] to continue or Ctrl-c to cancel adding it.
```

위의 명령어를 통해 저장소 목록에 mosquitto-dev PPA를 추가한다



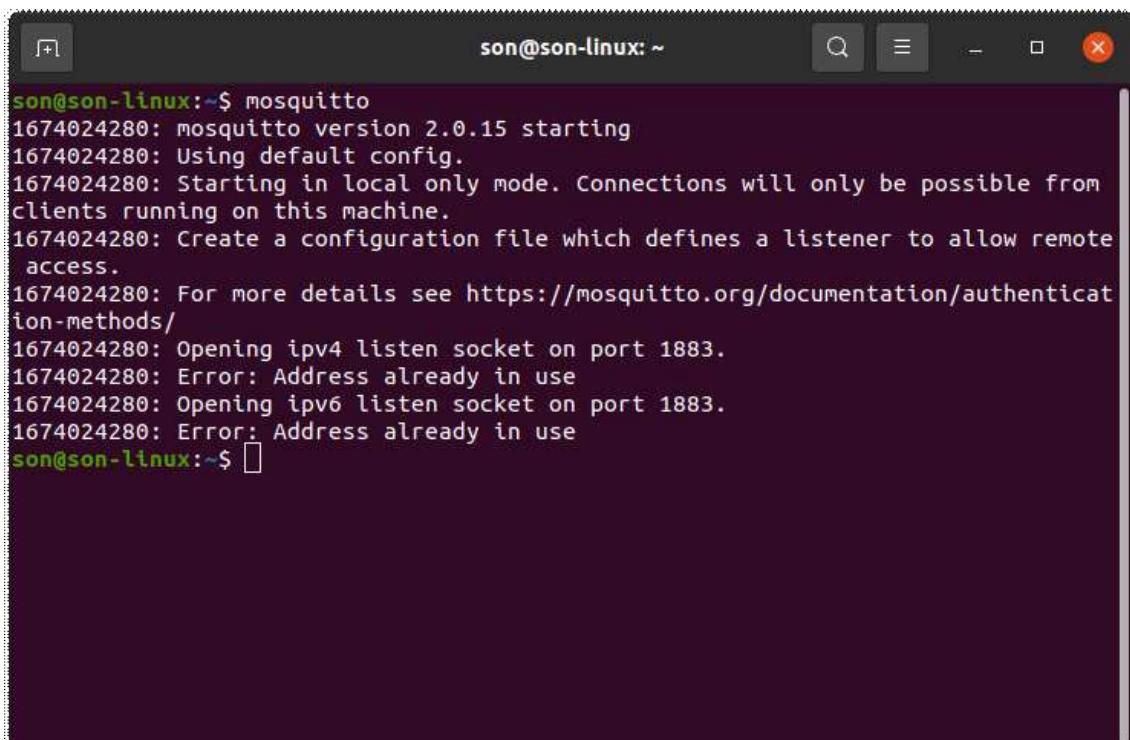
```
son@son-linux:~$ sudo apt-get update
Hit:1 https://deb.nodesource.com/node_16.x focal InRelease
Hit:2 http://repo.mysql.com/apt/ubuntu focal InRelease
Hit:3 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:5 http://ppa.launchpad.net/mosquitto-dev/mosquitto-ppa/ubuntu focal InRelease
Get:6 https://pkgs.tailscale.com/stable/ubuntu focal InRelease
Hit:7 http://kr.archive.ubuntu.com/ubuntu focal InRelease
Hit:8 http://kr.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:9 http://kr.archive.ubuntu.com/ubuntu focal-backports InRelease
Fetched 5,542 B in 2s (3,139 B/s)
Reading package lists... Done
son@son-linux:~$
```

mosquitto를 저장소에 추가한 최신 버전으로 업데이트한다.



```
son@son-linux:~$ sudo apt-get update
Hit:1 https://deb.nodesource.com/node_16.x focal InRelease
Hit:2 http://repo.mysql.com/apt/ubuntu focal InRelease
Hit:3 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:5 http://ppa.launchpad.net/mosquitto-dev/mosquitto-ppa/ubuntu focal InRelease
Get:6 https://pkgs.tailscale.com/stable/ubuntu focal InRelease
Hit:7 http://kr.archive.ubuntu.com/ubuntu focal InRelease
Hit:8 http://kr.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:9 http://kr.archive.ubuntu.com/ubuntu focal-backports InRelease
Fetched 5,542 B in 2s (3,139 B/s)
Reading package lists... Done
son@son-linux:~$ sudo apt-get install mosquitto -y
Reading package lists... done
Building dependency tree
Reading state information... done
The following packages were automatically installed and are no longer required:
  acl apg colord-data gnome-control-center-faces gnome-online-accounts
  hplip-data libcolord-gtk1 libcolorhug2 libdlt2 libev4 libgsound0
  libgupnp-av-1.0-2 libgupnp-dlna-2.0-3 libieee1284-3 libimagequant0
  librygel-core-2.6-2 librygel-db-2.6-2 librygel-renderer-2.6-2
  librygel-server-2.6-2 libsane-common libsnmp-base libwebsocket15
  mobile-broadband-provider-info network-manager-gnome
```

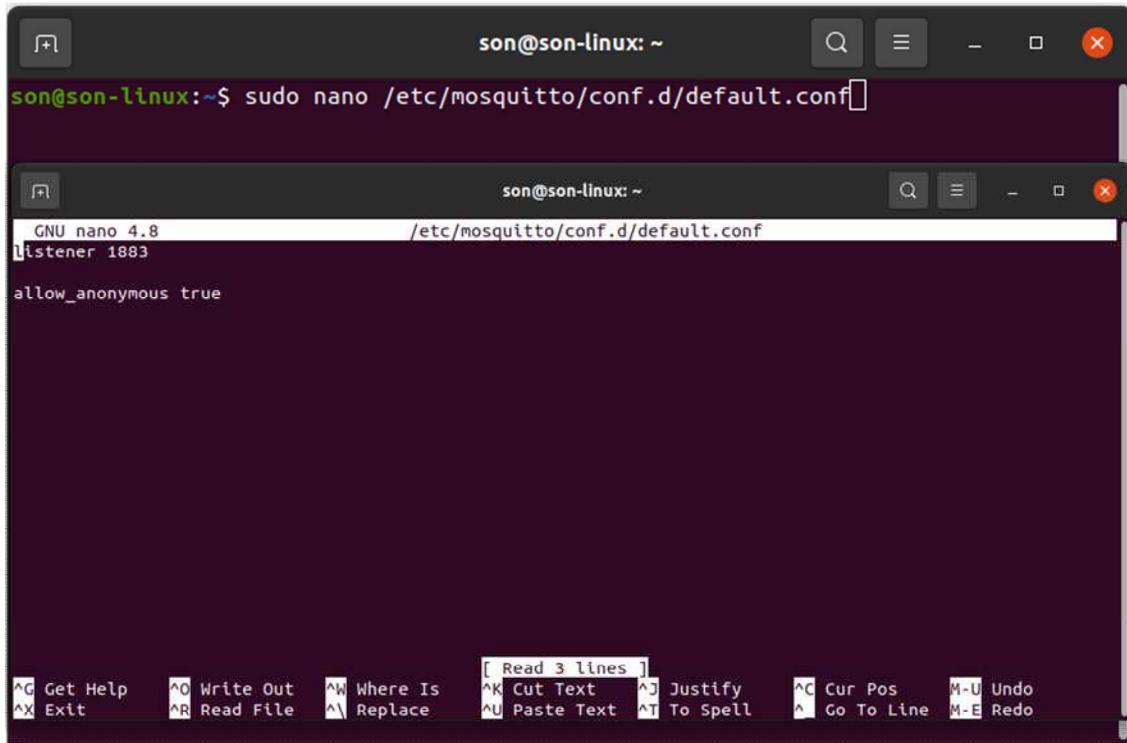
"sudo apt-get install mosquitto -y"을 통해 최신 버전 mosquitto를 설치한다.



```
son@son-linux:~$ mosquitto
1674024280: mosquitto version 2.0.15 starting
1674024280: Using default config.
1674024280: Starting in local only mode. Connections will only be possible from
clients running on this machine.
1674024280: Create a configuration file which defines a listener to allow remote
access.
1674024280: For more details see https://mosquitto.org/documentation/authentication-methods/
1674024280: Opening ipv4 listen socket on port 1883.
1674024280: Error: Address already in use
1674024280: Opening ipv6 listen socket on port 1883.
1674024280: Error: Address already in use
son@son-linux:~$
```

설치된 mosquitto 버전을 "mosquitto" 또는 "mosquitto -v" 명령어를 통해 확인한다.

mosquitto는 2.0 버전으로 업데이트 되면서 기존의 기본 설정을 사용하는 대신 사용자가 설정한 내용을 기반으로 동작한다. 이를 위해 아래와 같이 설정파일을 작성한다.

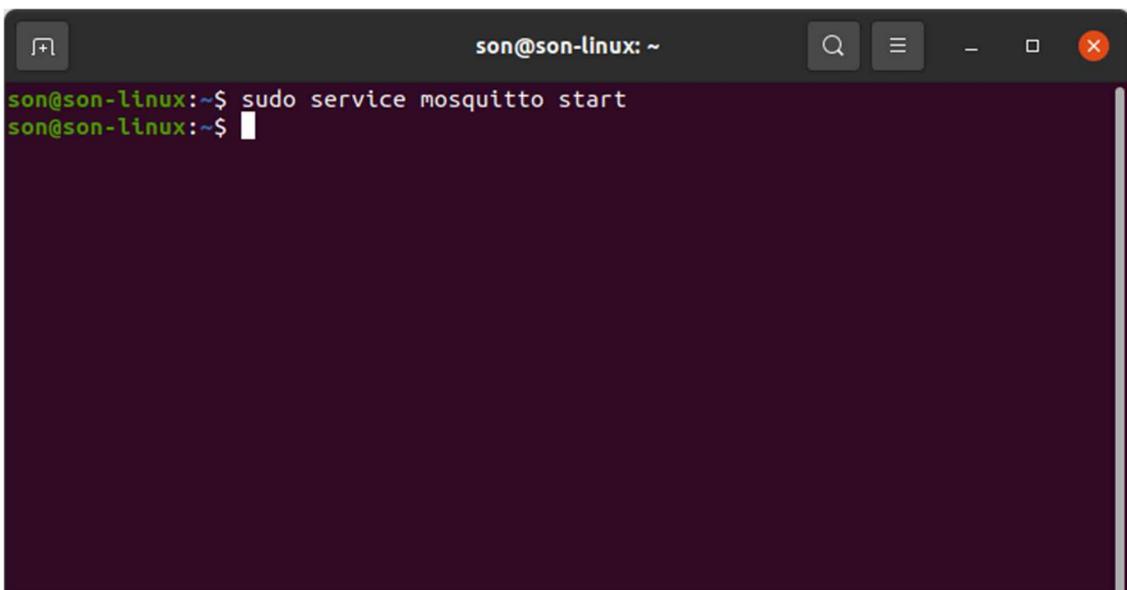


The screenshot shows two terminal windows. The top window displays the command: `son@son-linux:~$ sudo nano /etc/mosquitto/conf.d/default.conf`. The bottom window shows the contents of the configuration file:

```
GNU nano 4.8
listener 1883
allow_anonymous true
```

The terminal window has a dark background and light-colored text. The nano editor interface includes standard keyboard shortcuts at the bottom.

listener로 지정한 포트의 네트워크 연결을 수신한다. Mobius의 MQTT 포트인 1883 포트를 추가한다. 또한, allow\_anonymous true 설정을 통해 익명 클라이언트 연결을 허용한다.



The screenshot shows a terminal window with the command: `son@son-linux:~$ sudo service mosquitto start`. The output shows the service starting successfully.

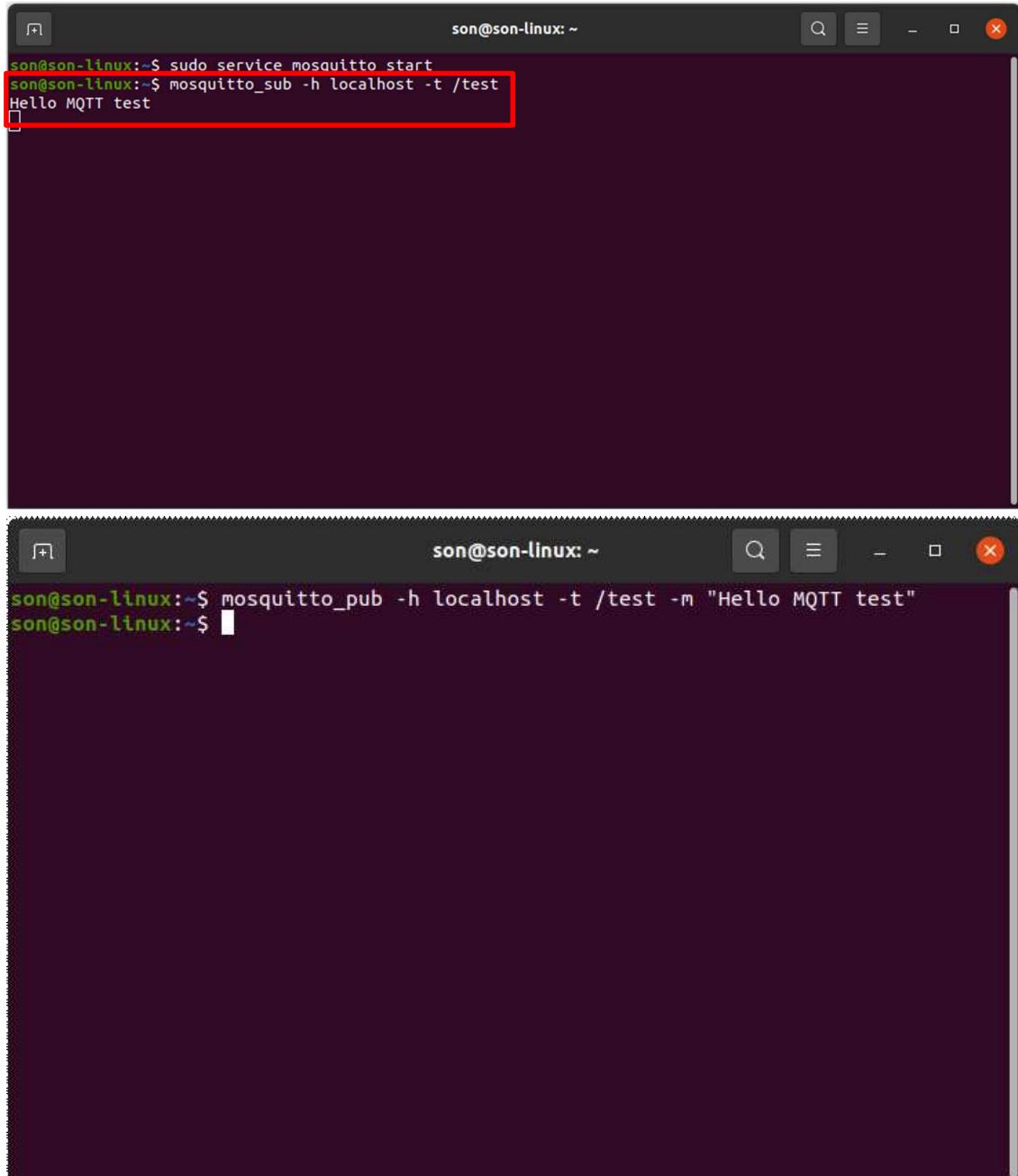
```
son@son-linux:~$ sudo service mosquitto start
son@son-linux:~$
```

mosquitto의 변경된 설정 반영을 위해 mosquitto 서비스를 재실행한다.

다음 과정을 통해 설치된 mosquitto를 동작테스트를 할 수 있다.

```
$ mosquitto_sub -h localhost -t /mytopic/1
```

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



The image displays two terminal windows side-by-side, both titled 'son@son-linux: ~'. The top window shows the command 'mosquitto\_sub -h localhost -t /test' being run, with the output 'Hello MQTT test' highlighted by a red box. The bottom window shows the command 'mosquitto\_pub -h localhost -t /test -m "Hello MQTT test"' being run.

```
son@son-linux:~$ sudo service mosquitto start
son@son-linux:~$ mosquitto_sub -h localhost -t /test
Hello MQTT test
```

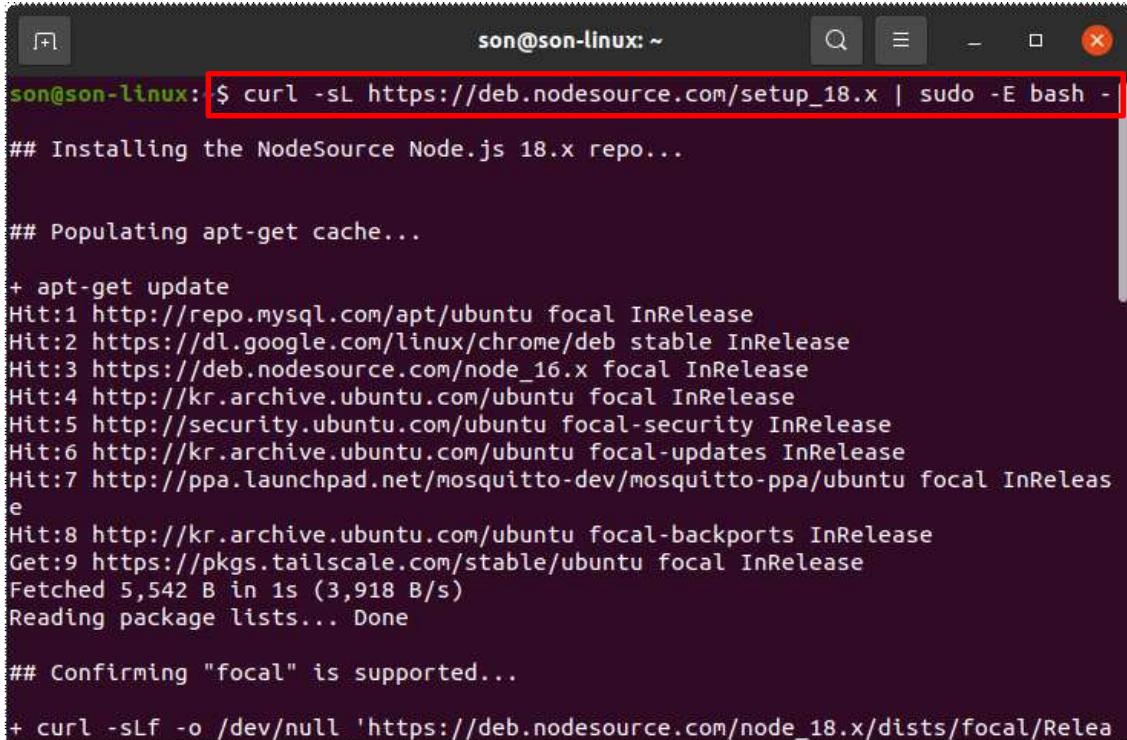
  

```
son@son-linux:~$ mosquitto_pub -h localhost -t /test -m "Hello MQTT test"
son@son-linux:~$
```

### 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 모듈

을 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.



The screenshot shows a terminal window titled "son@son-linux: ~". The command entered is "curl -sL https://deb.nodesource.com/setup\_18.x | sudo -E bash -". The output of the command is displayed below, showing the process of installing the NodeSource Node.js 18.x repository, updating the apt-get cache, and confirming that the "focal" release is supported. The entire command line and its output are highlighted with a red box.

```
son@son-linux: $ curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
## Installing the NodeSource Node.js 18.x repo...
## Populating apt-get cache...
+ apt-get update
Hit:1 http://repo.mysql.com/apt/ubuntu focal InRelease
Hit:2 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:3 https://deb.nodesource.com/node_16.x focal InRelease
Hit:4 http://kr.archive.ubuntu.com/ubuntu focal InRelease
Hit:5 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:6 http://kr.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:7 http://ppa.launchpad.net/mosquitto-dev/mosquitto-ppa/ubuntu focal InRelease
Hit:8 http://kr.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:9 https://pkgs.tailscale.com/stable/ubuntu focal InRelease
Fetched 5,542 B in 1s (3,918 B/s)
Reading package lists... Done
## Confirming "focal" is supported...
+ curl -sLf -o /dev/null 'https://deb.nodesource.com/node_18.x/dists/focal/Relea
```

Node.js v18.x 버전은 패키지에서 바로 설치하는 것이 불가능하기 때문에 데비안과 우분투 운영체제의 바이너리 배포판 저장소 Node source 홈페이지에서 제공하는 최신 v18 LTS를 "curl -sL https://deb.nodesource.com/setup\_18.x | sudo -E bash -" 명령어를 통해 apt 저장소를 추가한다.

```

son@son-linux:~$ sudo apt-get install -y nodejs
Reading package lists... done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  acl apg colord-data gnome-control-center-faces gnome-online-accounts
  hplip-data libcolorerd-gtk1 libcolorhug2 libdlt2 libev4 libgsound0
  libgupnp-av-1.0-2 libgupnp-dlna-2.0-3 libieee1284-3 libimagequant0
  librygel-core-2.6-2 librygel-db-2.6-2 librygel-renderer-2.6-2
  librygel-server-2.6-2 libsane-common libsnmp-base libwebsockets15
  mobile-broadband-provider-info network-manager-gnome
  printer-driver-postscript-hp python3-macaroonbakery python3-olefile
  python3-pil python3-protobuf python3-pymacaroons python3-renderpm
  python3-reportlab python3-reportlab-accel python3-rfc3339 python3-tz rygel
Use 'sudo apt autoremove' to remove them.
The following packages will be upgraded:
  nodejs
1 upgraded, 0 newly installed, 0 to remove and 74 not upgraded.
Need to get 28.4 MB of archives.
After this operation, 56.0 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_18.x focal/main amd64 nodejs amd64 18.13.0
-deb-1nodesource1 [28.4 MB]
Fetched 28.4 MB in 4s (7,174 kB/s)
(Reading database ... 198211 files and directories currently installed.)

```

저장소 추가 후 설명된 Node.js 설치 명령어 “sudo apt-get install -y nodejs”를 통해 최신 버전 Node.js를 설치한다.

```

son@son-linux:~$ sudo apt-get install -y nodejs
Reading package lists... done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  acl apg colord-data gnome-control-center-faces gnome-online-accounts
  hplip-data libcolorerd-gtk1 libcolorhug2 libdlt2 libev4 libgsound0
  libgupnp-av-1.0-2 libgupnp-dlna-2.0-3 libieee1284-3 libimagequant0
  librygel-core-2.6-2 librygel-db-2.6-2 librygel-renderer-2.6-2
  librygel-server-2.6-2 libsane-common libsnmp-base libwebsockets15
  mobile-broadband-provider-info network-manager-gnome
  printer-driver-postscript-hp python3-macaroonbakery python3-olefile
  python3-pil python3-protobuf python3-pymacaroons python3-renderpm
  python3-reportlab python3-reportlab-accel python3-rfc3339 python3-tz rygel
Use 'sudo apt autoremove' to remove them.
The following packages will be upgraded:
  nodejs
1 upgraded, 0 newly installed, 0 to remove and 74 not upgraded.
Need to get 28.4 MB of archives.
After this operation, 56.0 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_18.x focal/main amd64 nodejs amd64 18.13.0
-deb-1nodesource1 [28.4 MB]
Fetched 28.4 MB in 4s (7,174 kB/s)
(Reading database ... 198211 files and directories currently installed.)
Preparing to unpack .../nodejs_18.13.0-deb-1nodesource1_amd64.deb ...
Unpacking nodejs (18.13.0-deb-1nodesource1) over (16.18.0-deb-1nodesource1) ...
Setting up nodejs (18.13.0-deb-1nodesource1) ...
Processing triggers for man-db (2.9.1-1) ...
son@son-linux:~$ node -v
v18.13.0
son@son-linux:~$ 

```

“node -v” 명령어를 통해 설치한 버전을 확인한다.

## 2.3. Mobius 설치

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

### 2.3.1. 다운로드 및 모듈 설치

Mobius는 OCEAN Alliance에서 제공하는 GitHub (<https://github.com/IoTKETI/Mobius>)에 공개되어 있다.

The screenshot shows the GitHub repository page for `IoTKEITI/Mobius`. The repository is public and has 449 commits. The commit history lists several updates, including fixes for certification tests and optimizations. The repository page also displays project details such as oneM2M IoT Server Platform, nodejs, iot, server, mobius, and onem2m tags, a README file, BSD-3-Clause license, 112 stars, 19 watching, and 81 forks. It also includes sections for Releases, Packages, and Contributors.

Commit	Message	Author	Date
gbsmfather - fixed error (Re-certification test for updated Mobius)	update document of installation Guide at doc directory	es82521	2 weeks ago
doc	- fixed error (Re-certification test for updated Mobius)		8 months ago
mobius	- update sgn (timeout)		2 weeks ago
.gitignore	optimize code		3 years ago
LICENSE	- to unuse sri table of db when retrieve resource for check resour...		5 years ago
README.md	- fixed error (Re-certification test for updated Mobius)		2 years ago
app.js	git flow start		2 weeks ago
ca-crt.pem	- add cors		5 years ago
mobxus.js	- optimize code removing unused codes		last year
package.json	- add sgn_man		4 years ago
pxy_coap.js	- continue to update code for memory leak of node js		3 years ago
pxy_mqtt.js	git flow start		5 years ago
pxy_ws.js	- optimize code for memory leak of node js		5 years ago
server-crt.pem	- optimize code for memory leak of node js		5 years ago
server-key.pem	git flow start		5 years ago
wdt.js	optimize code		5 years ago



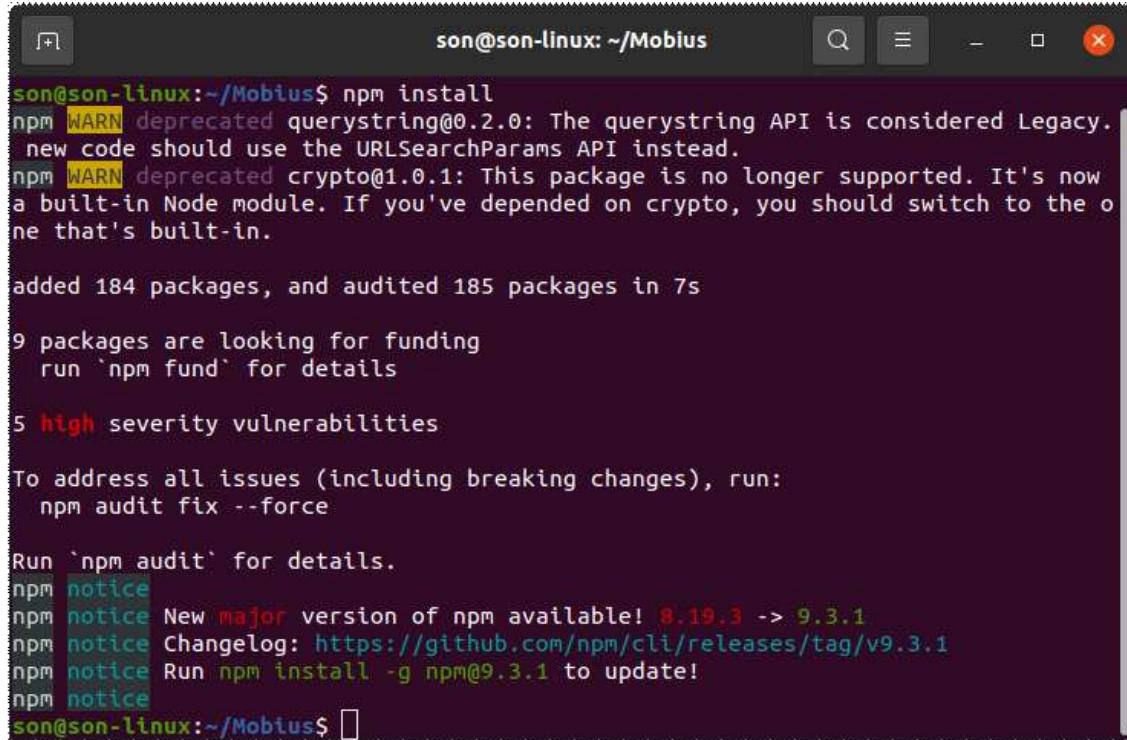
mobius.js가 Mobius의 메인 코드를 실행하는 파일이다. app.js는 대문 역할과 라우팅 역할을 수행하는 메인 코드이다. mobius 디렉토리에 있는 파일들이 실제 패킷을 처리하고 DB 액세스와 응답을 수행하는 코드들이다. 현재는 추가적인 node.js 모듈들이 설치가 되어있지 않아 에러가 날 것이다. 다음과 같이 명령 프롬프트를 실행시키고 npm 툴을 이용해 필요한 모듈을 설치해야 한다. package.json 파일에 필요한 모듈 리스트를 가지고 있다.

GitHub에 공개된 소스코드를 다운로드하는 여러 방법이 있지만 여기선 CLI 명령어를 통해 다운로드한다.

A screenshot of a terminal window titled "son@son-linux: ~". The command "git clone https://github.com/IoTKETI/Mobius.git" is being executed. The output shows the progress of cloning the repository, including object enumeration, counting, compressing, and receiving objects, and finally resolving deltas. The process is completed successfully.

```
son@son-linux: $ git clone https://github.com/IoTKETI/Mobius.git
Cloning into 'Mobius'...
remote: Enumerating objects: 2454, done.
remote: Counting objects: 100% (187/187), done.
remote: Compressing objects: 100% (103/103), done.
remote: Total 2454 (delta 119), reused 138 (delta 84), pack-reused 2267
Receiving objects: 100% (2454/2454), 43.82 MiB | 9.72 MiB/s, done.
Resolving deltas: 100% (1864/1864), done.
son@son-linux:~$
```

"git clone <https://github.com/IoTKETI/Mobius.git>" 명령어를 통해 Mobius 소스코드를 다운로드한다. 이후 "cd Mobius" 명령어를 통해 다운로드 한 Mobius 폴더로 이동한다.



```
son@son-linux:~/Mobius$ npm install
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy.
new code should use the URLSearchParams API instead.
npm WARN deprecated crypto@1.0.1: 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.

added 184 packages, and audited 185 packages in 7s

9 packages are looking for funding
  run `npm fund` for details

5 high severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

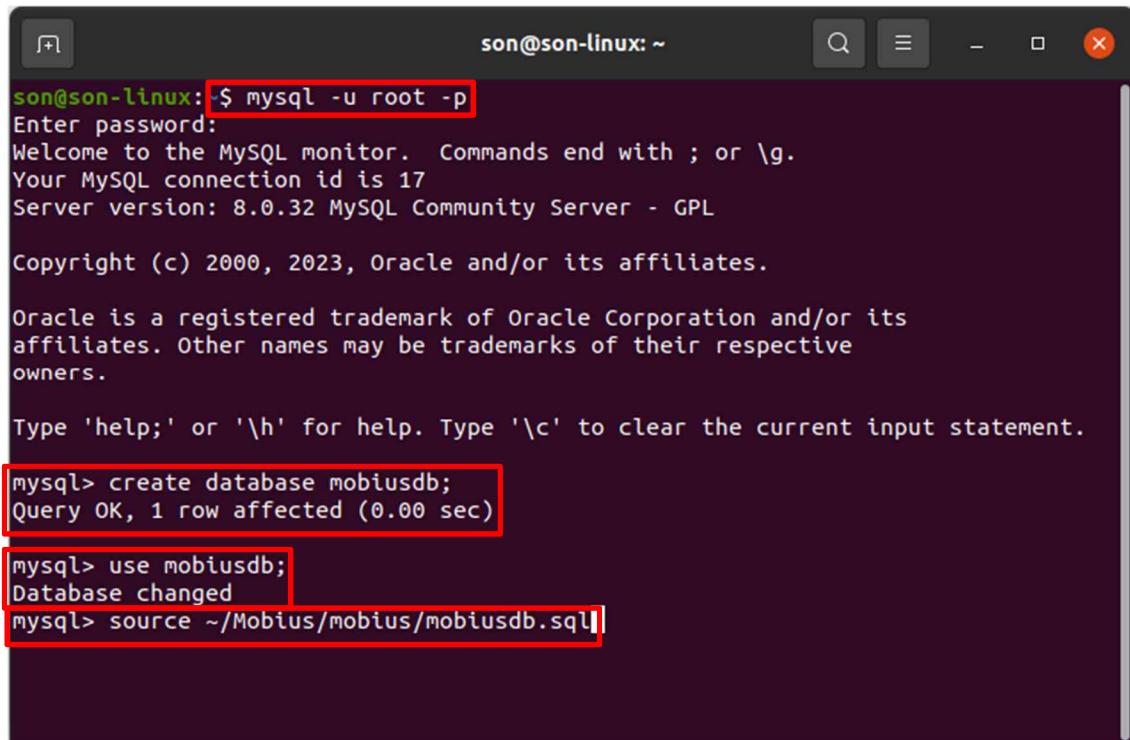
Run `npm audit` for details.
npm notice
npm notice New major version of npm available! 8.19.3 -> 9.3.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v9.3.1
npm notice Run npm install -g npm@9.3.1 to update!
npm notice
son@son-linux:~/Mobius$
```

위와 같이 “npm install”을 입력하여 Mobius 동작에 필요한 모듈들을 설치한다.

설치가 완료되면 node\_modules 폴더가 생긴 것을 확인할 수 있다.

### 2.3.2. mobiusdb 데이터베이스 생성(Linux)

Mobius 소스코드 폴더 안에 MySQL 데이터베이스 구성을 위한 mobiusdb.sql 파일이 있다. 이 파일을 통해 데이터베이스를 구성한다.



```
son@son-linux: ~ $ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.32 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

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.

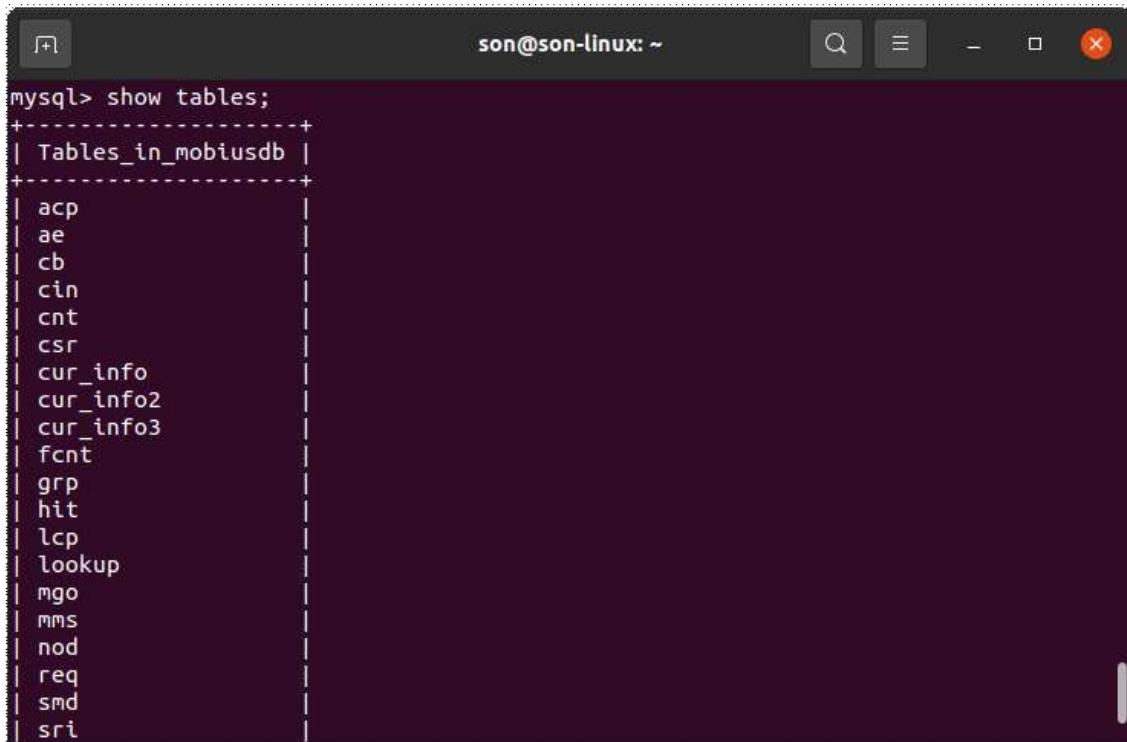
① mysql> create database mobiusdb;
Query OK, 1 row affected (0.00 sec)

② mysql> use mobiusdb;
Database changed

③ mysql> source ~/Mobius/mobius/mobiusdb.sql
```

먼저 “mysql -u root -p” 명령어를 입력하여 root로 로그인 한다. 입력하면 패스워드를 입력하라는 메시지를 볼 수 있는데 MySQL을 설치할 때 생성했던 패스워드를 입력한다. Mobius와 연동을 위한 MySQL 설정은 다음과 같은 순서로 SQL문 입력을 통해 진행한다.

- ①. 데이터베이스 생성 - “CREATE DATABASE mobiusdb;”
- ②. 생성한 데이터베이스 선택 - “USE mobiusdb;”
- ③. mobiusdb.sql 파일을 통한 Mobius 테이블 생성 - “source ~/Mobius/mobius/mobiusdb.sql”



```
son@son-linux: ~
mysql> show tables;
+-----+
| Tables_in_mobiusdb |
+-----+
| acp
| ae
| cb
| cin
| cnt
| csr
| cur_info
| cur_info2
| cur_info3
| fcnt
| grp
| hit
| lcp
| lookup
| mgo
| mms
| nod
| req
| smd
| sri
+-----+
```

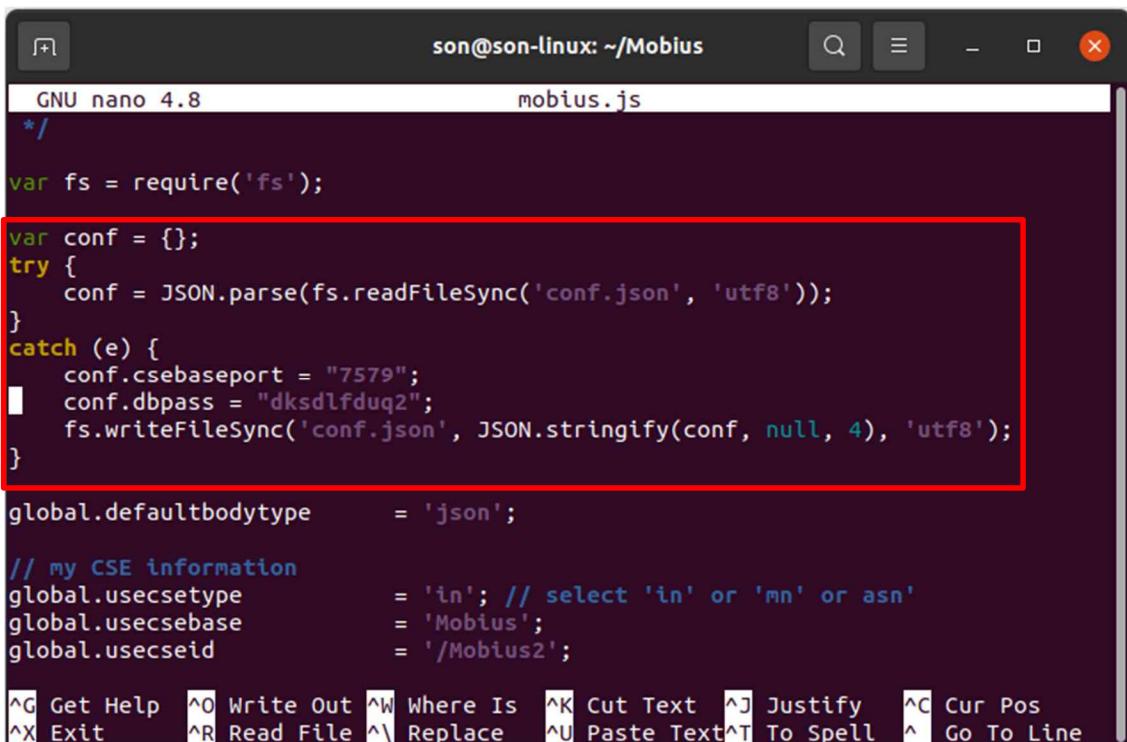
DB에 정상적으로 테이블이 생성되었는지 확인을 위해 "SHOW TABLES;"를 통해 테이블이 구성되어있는 것을 확인 수 있다.

## 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 mobius.js 파일을 통해 설정을 변경할 수 있다.



```

GNU nano 4.8          son@son-linux: ~/Mobius
*/



var fs = require('fs');

var conf = {};
try {
    conf = JSON.parse(fs.readFileSync('conf.json', 'utf8'));
}
catch (e) {
    conf.csebaseport = "7579";
    conf.dbpass = "dksdlfdq2";
    fs.writeFileSync('conf.json', JSON.stringify(conf, null, 4), 'utf8');
}

global.defaultbodytype      = 'json';

// my CSE information
global.usecsetype           = 'in'; // select 'in' or 'mn' or 'asn'
global.usecsebase            = 'Mobius';
global.usecseid              = '/Mobius2';

^G Get Help   ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace   ^U Paste Text^T To Spell  ^_ Go To Line

```

Mobius 구동 시 필요한 설정은 위와 같이 mobius.js 또는 conf.json에서 Mobius가 오픈하는 포트 번호와 환경 구축 시 설치한 MySQL 데이터베이스 접근 암호(dbpass)를 설정 해줘야 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.

```

son@son-linux:~/Mobius$ node mobius.js
Production Mode
Garbage collection is not exposed
sgn_mqtt_client is connected
CPU Count: 16
select_ri_lookup /Mobius: 5.088ms
update_cb_poa_csi /Mobius: 1.714ms
"""

Production Mode
Garbage collection is not exposed
sgn_mqtt_client is connected
Production Mode
Garbage collection is not exposed
sgn_mqtt_client is connected
mobius server (192.168.50.74) running at 7579 port
select_ri_lookup /Mobius: 1.702ms
Production Mode
Garbage collection is not exposed
update_cb_poa_csi /Mobius: 0.859ms
"""

mobius server (192.168.50.74) running at 7579 port
sgn_mqtt_client is connected
select_ri_lookup /Mobius: 2.069ms
update_cb_poa_csi /Mobius: 1.084ms

```

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.defaultnmtype      = '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.usepxymqttport        = '7578';

global.usetsagentport        = '7582';

global.usemqttbroker         = 'localhost'; // mqttbroker for mobius
global.useselectorbroker     = '10.10.202.114';
global.usesecure              = 'disable';
if(useselect == '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

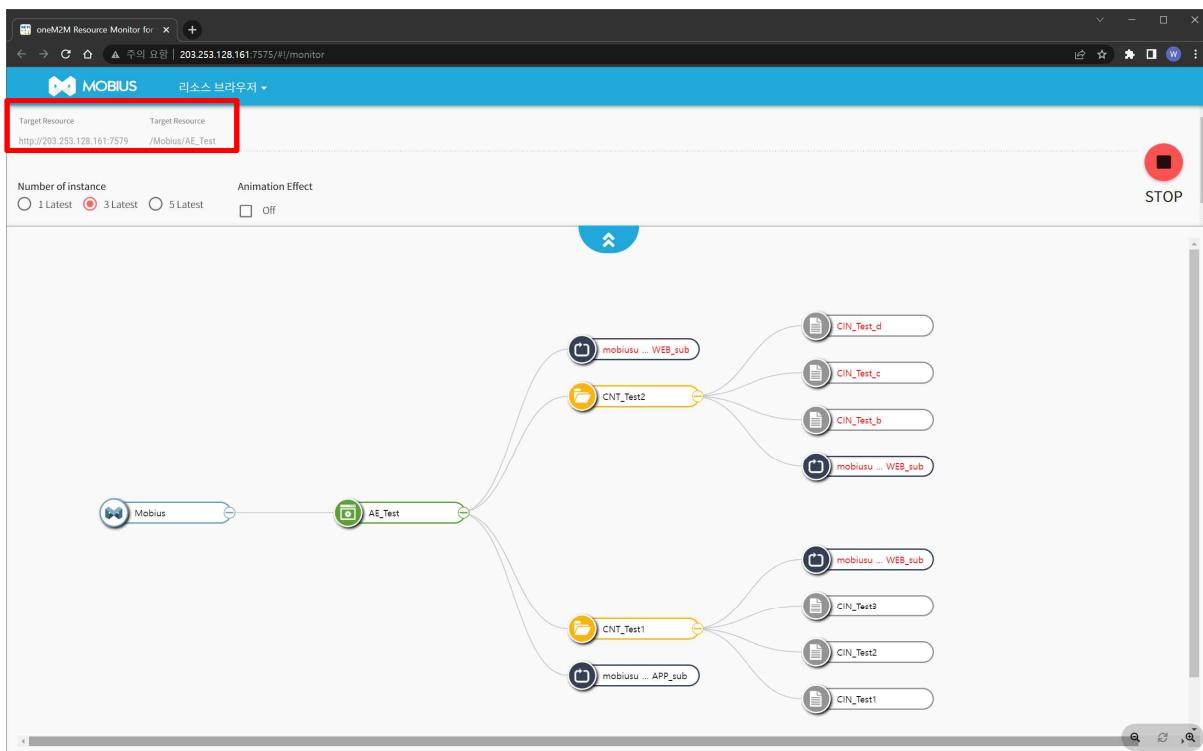
### 3.2.1. 리소스뷰어(oneM2M Browser) 활용

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>

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

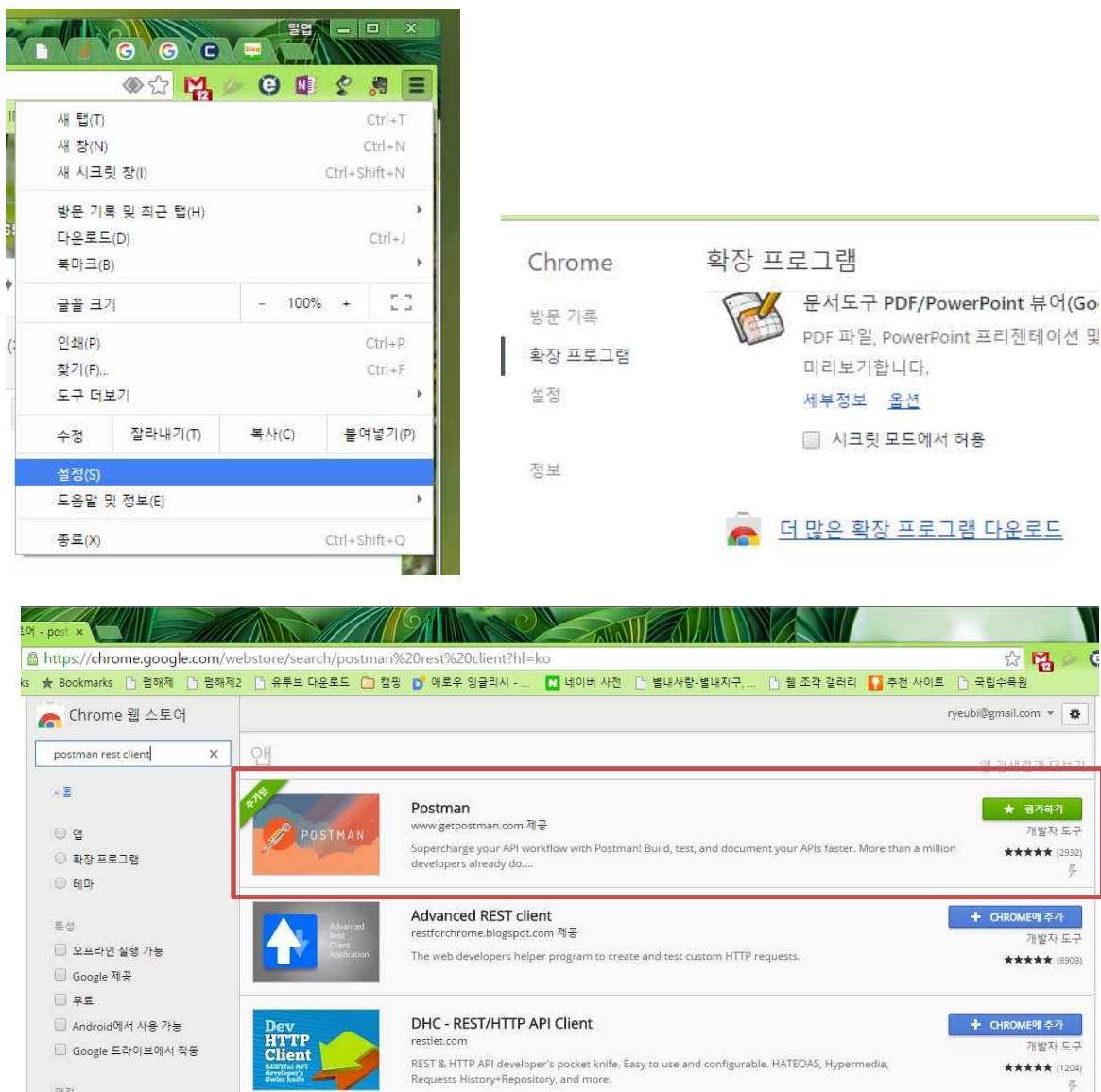


아래와 같이 리소스뷰어가 보이면 표시된 부분에 Mobius에 생성된 리소스의 url을 입력하고 Strat 버튼을 누르면 해당 url의 리소스 구조를 볼 수 있다. url 부분의 처음은 콤보 박스로 되어 있고 이전 Mobius와 Mobius release 2를 선택할 수 있다. 이후 부분은 직접 입력해야 하고 여기에 생성한 AE의 name을 입력하면 된다.

### 3.2.2. Postman 활용

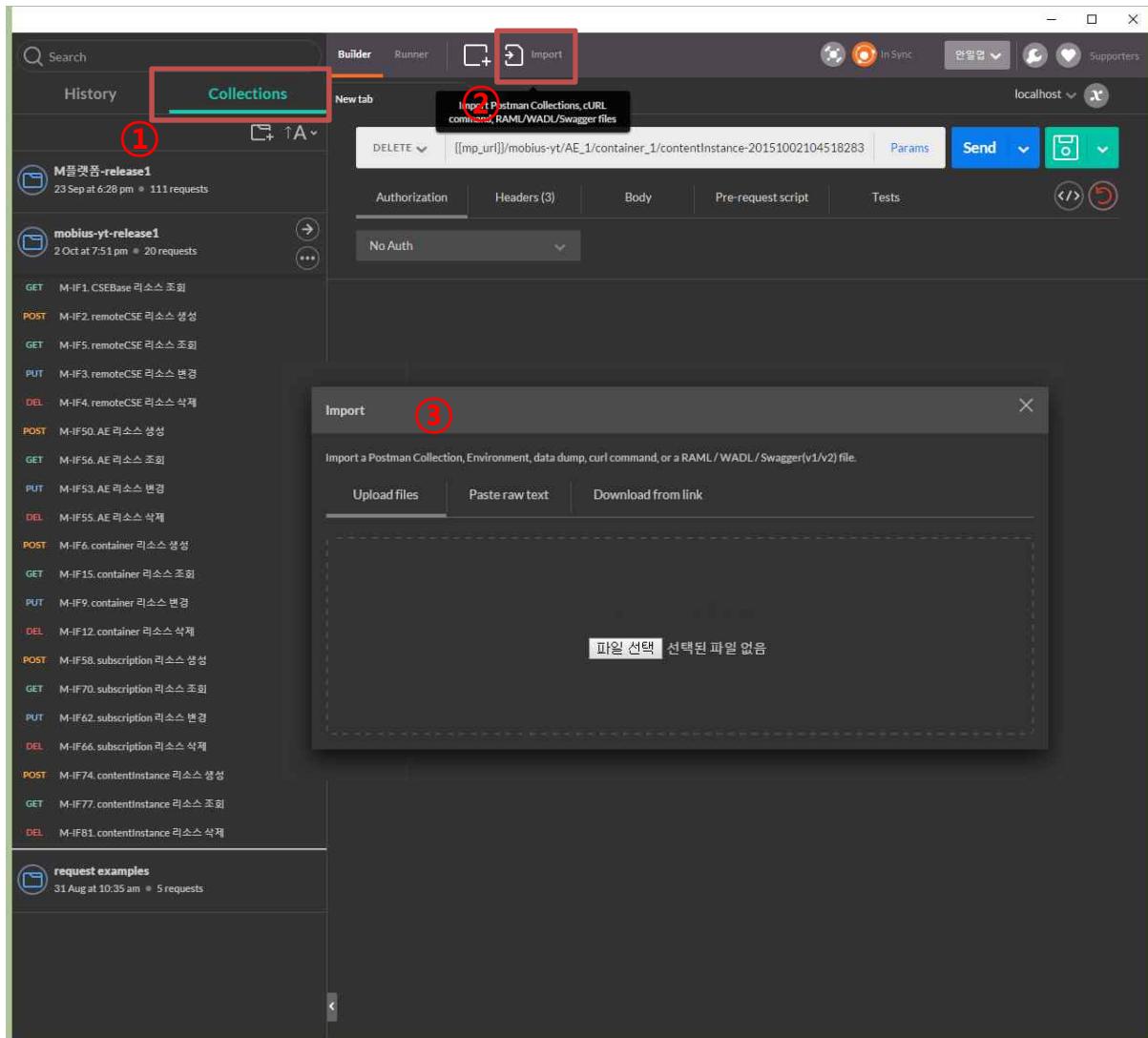
크롬 확장 프로그램인 Postman을 활용하여 직접 구축한 Mobius를 동작을 테스트 할 수 있다.

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 창

이 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 the Mobius API Explorer interface. At the top, it displays the project name "mobius-xt-release1" and the date "2 Oct at 7:51 pm" with "20 requests". Below this, a list of 20 API requests is shown:

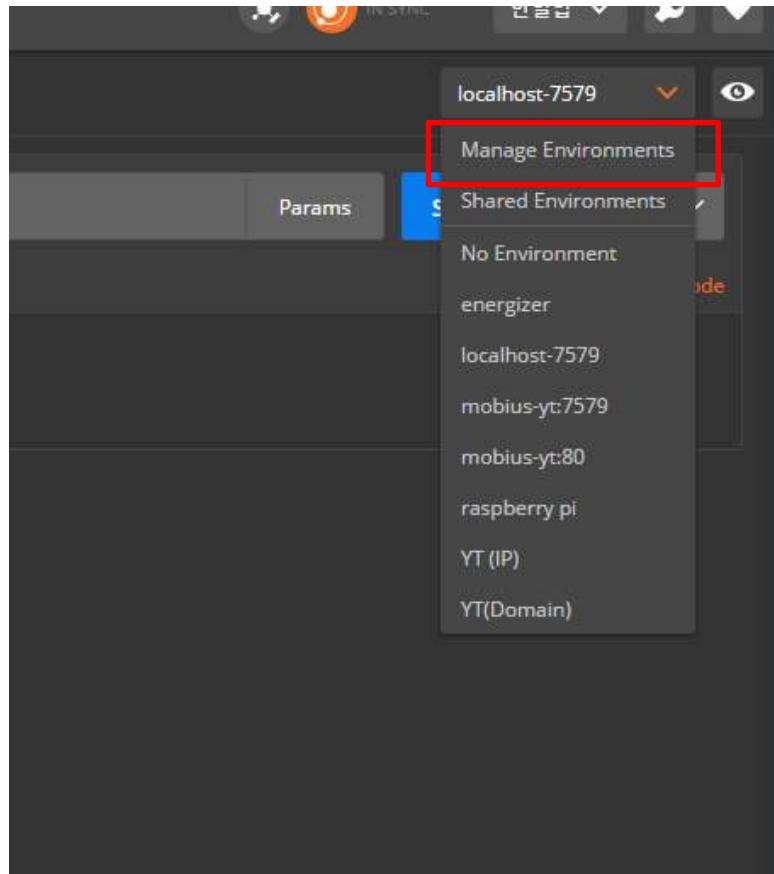
- GET** M-IF1. CSEBase 리소스 조회
- POST** M-IF2. remoteCSE 리소스 생성
- GET** M-IF5. remoteCSE 리소스 조회
- PUT** M-IF3. remoteCSE 리소스 변경
- DEL** M-IF4. remoteCSE 리소스 삭제
- POST** M-IF50. AE 리소스 생성
- GET** M-IF56. AE 리소스 조회
- PUT** M-IF53. AE 리소스 변경
- DEL** M-IF55. AE 리소스 삭제
- POST** M-IF6. container 리소스 생성
- GET** M-IF15. container 리소스 조회
- PUT** M-IF9. container 리소스 변경
- DEL** M-IF12. container 리소스 삭제
- POST** M-IF58. subscription 리소스 생성
- GET** M-IF70. subscription 리소스 조회
- PUT** M-IF62. subscription 리소스 변경
- DEL** M-IF66. subscription 리소스 삭제
- POST** M-IF74. contentInstance 리소스 생성
- GET** M-IF77. contentInstance 리소스 조회
- DEL** M-IF81. contentInstance 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.

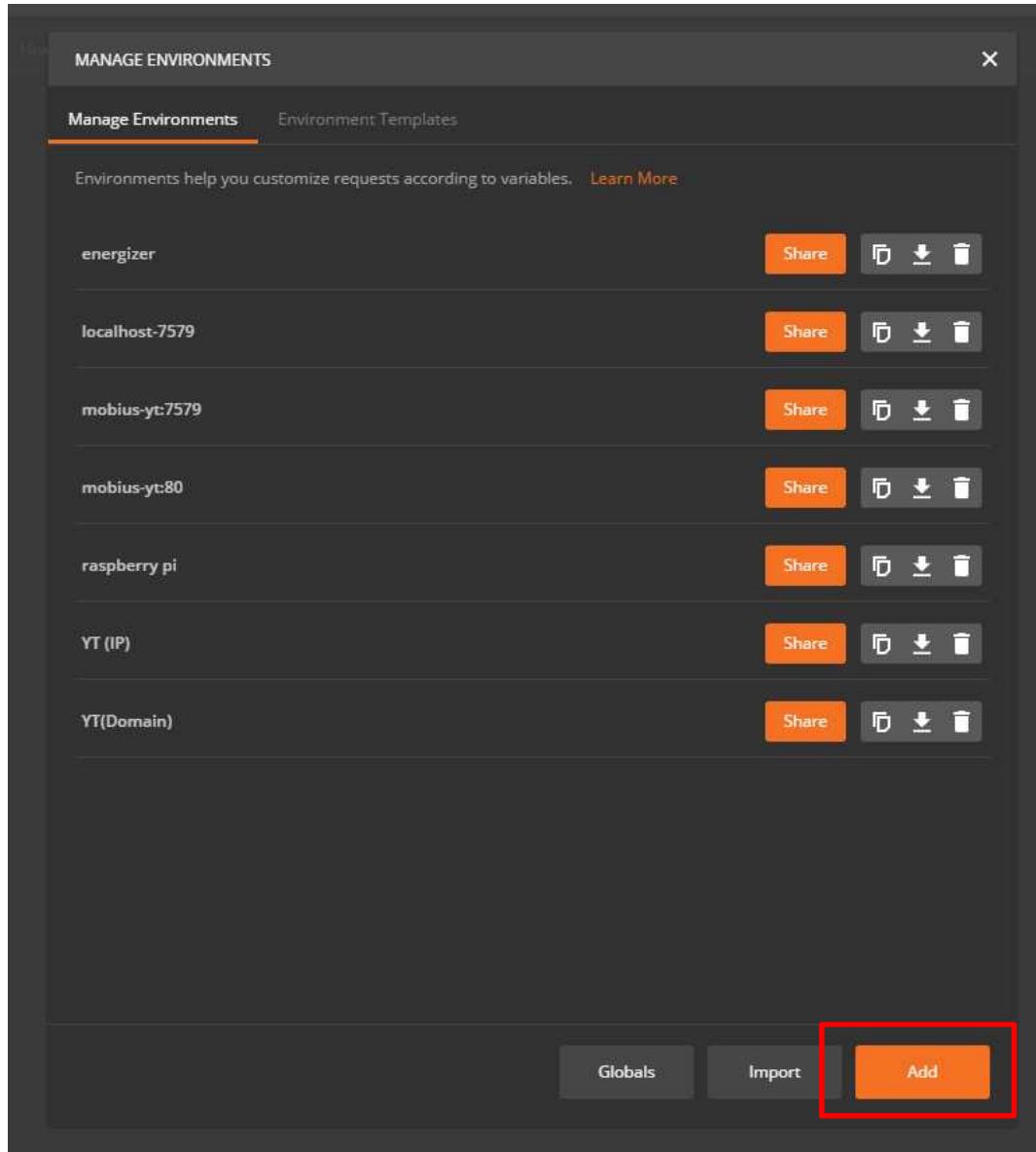
The screenshot shows the Postman application interface. On the left, a sidebar lists projects and collections. In the main area, a specific request is being configured:

- M-IF1. CSEBase 리소스 조회**
- Method:** GET
- URL:** http://203.253.174.104:7579/mobius-xt
- Authorization:** No Auth
- Headers (4):** None
- Body:** None
- Pre-request script:** None
- Tests:** None

그러면, {{mp\_url}} 설정하는 방법은 아래와 같다.



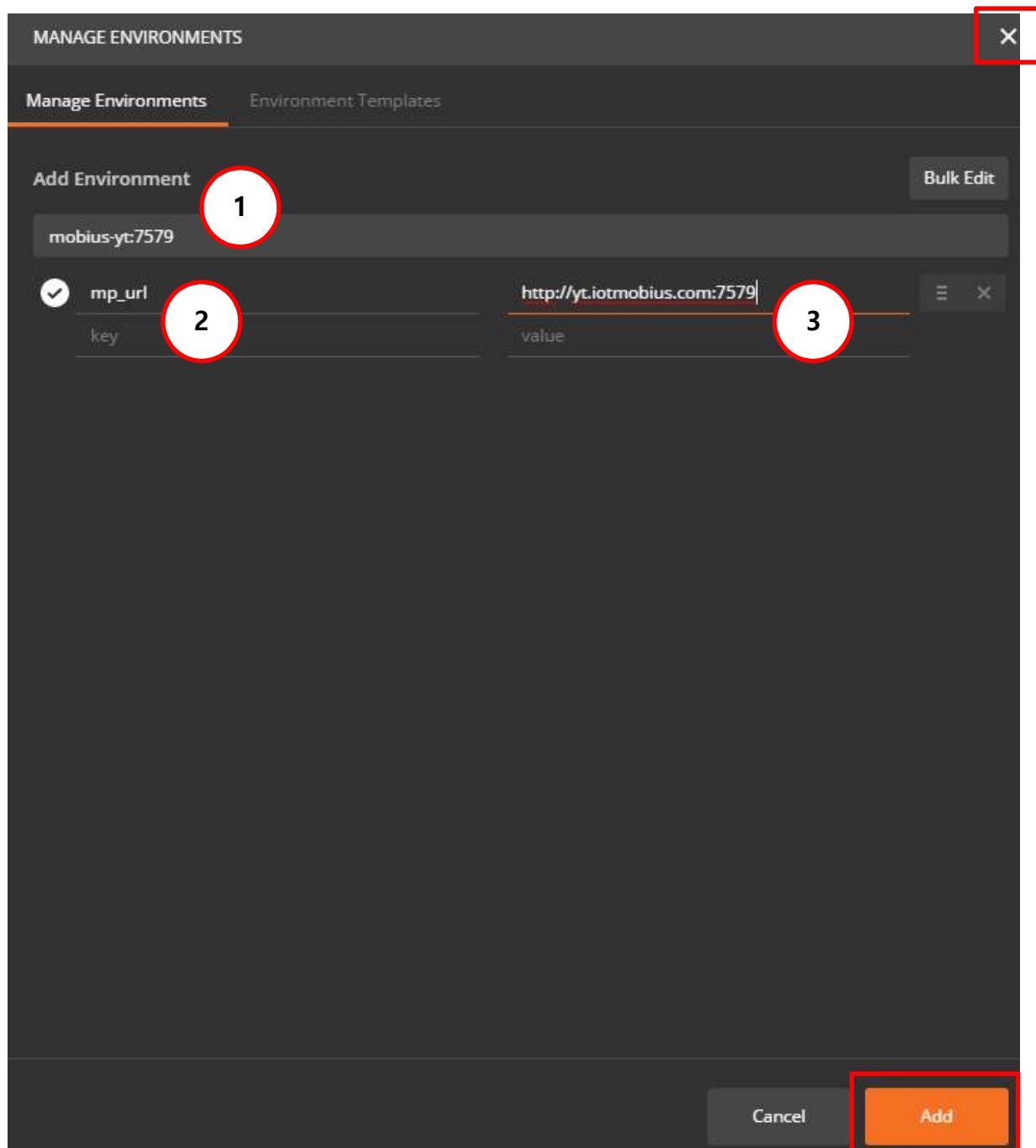
포스트맨 창에서 오른쪽 위에 보면 설정할 수 있는 리스트박스가 있고 이를 선택하면 그림과 같이 리스트 메뉴가 나온다. 이 때 Manage Environments 를 선택한다.



설정 창이 나오면 Add 버튼을 눌러 아래와 같이 {{mp\_url}} 값을 입력한다.

- ① 설정에 대한 이름을 입력한다. (mobius-yt: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.