



Mobius Release 2

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	2022-01-12	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 추가될 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 플랫폼의 소스코드는 하기의 라이선스 정책에 따라 배포됩니다.

- OCEAN 기반 오픈 소스코드는 3-clause BSD-style 라이선스를 기반으로 배포한다. 3-clause BSD-style 라이선스에 따라, 소스코드 파일에 저작권 헤더를 유지하는 동안 OCEAN 기반 오픈 소스 코드는 상업용 또는 비상업용 시스템에 자유롭게 사용할 수 있다.
- OCEAN 기반 오픈 소스 라이선스는 개발된 소스코드를 다른 사람과 공유하는 것을 요구하지는 않는다. OCEAN 기반 오픈 소스를 활용하여 개발한 소스코드는 개발자의 것이고 누구와 공유할 의무는 없다.
- 누구나 OCEAN 의 오픈 소스 발전에 공헌할 수 있다. 그러나, 개발된 소스코드가 OCEAN 에 공헌되기를 원한다면 이 코드는 OCEAN 의 라이선스 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. Mobius 서버 플랫폼 개요.....	7
1.2.2. Mobius 서버 플랫폼 주요 기능.....	8
1.2.3. Mobius 서버 플랫폼 구성.....	8
1.2.4. Mobius 서버 플랫폼 S/W 아키텍처.....	9
1.2.5. Mobius 서버 디렉토리 구조.....	11
2. Mobius 서버 플랫폼 설치.....	15
2.1. 설치 개요.....	15
2.2. Mobius 구동환경 구축 (Windows).....	15
2.2.1. MySQL 설치.....	15
2.2.1. MQTT 브로커 설치.....	20
2.2.2. Node JS 설치.....	24
2.3. Mobius 설치.....	27
2.3.1. 다운로드 및 모듈 설치.....	27
2.3.2. mobiusdb 데이터베이스 생성.....	30
3. Mobius 서버 플랫폼 구동.....	32
3.1. 환경설정.....	32
3.2. 테스트.....	34
3.2.1. 리소스뷰어(oneM2M Browser) 활용.....	34
3.2.2. Postman 활용.....	37

1. Mobius 소개

1.1. Mobius 란?

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.

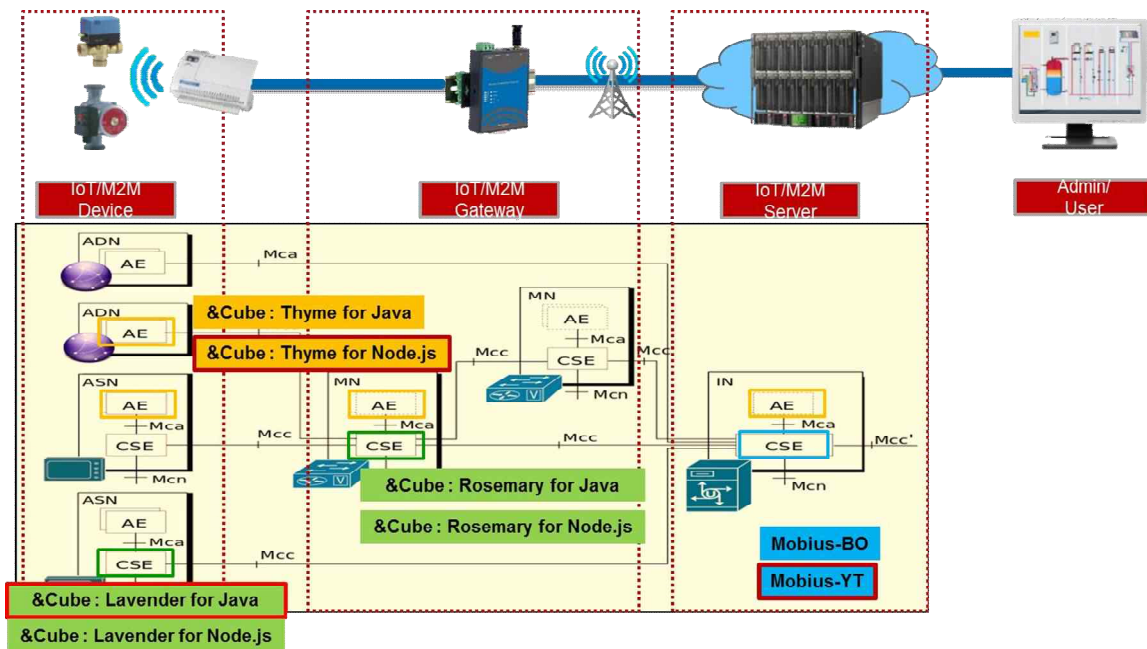


Figure 1 oneM2M 기반 Mobius 플랫폼 개념도

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

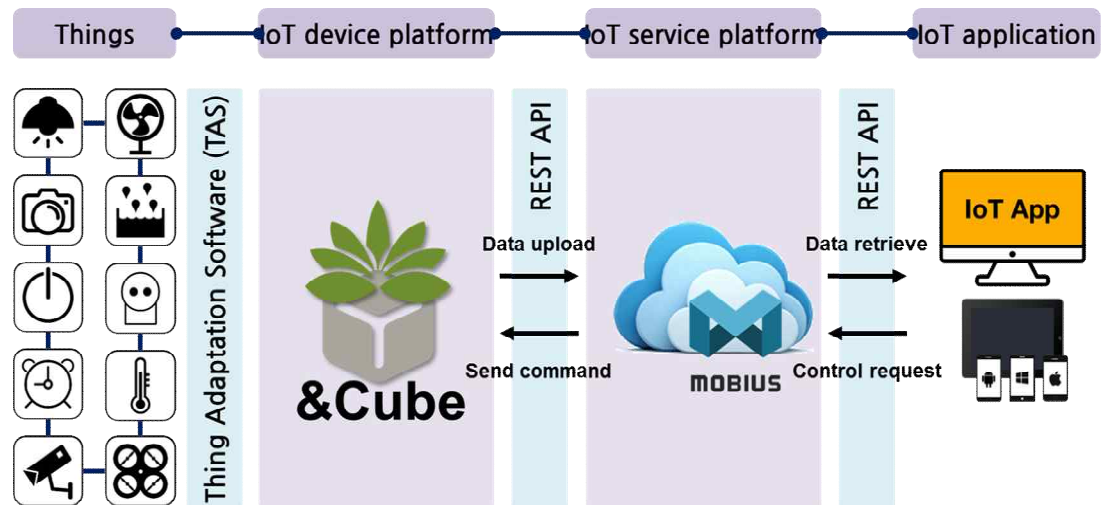


Figure 2 Mobius 와 IoT 디바이스 연동

Mobius 플랫폼은 리소스를 생성하면 자동으로 해당 리소스를 접근할 수 있는 API 가 자동으로 생성되며 이를 위한 인터페이스로 REST API 인 OPEN API 를 제공한다.

1.2. Mobius 플랫폼

1.2.1. Mobius 서버 플랫폼 개요

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

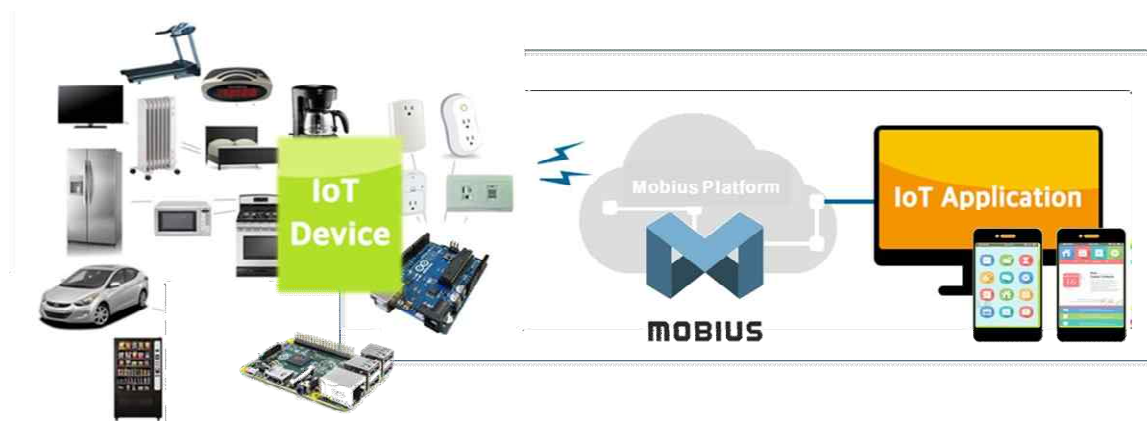


Figure 3 Mobius Platform 개요

Mobius 는 Node.js 를 사용하여 JavaScript 언어로 개발된 Mobius 서버 플랫폼으로 국제 표준인 oneM2M 표준에 따라 개발되었다. Mobius 는 Node.js 의 express 모듈을 사용하여 개발되었다.

Mobius Platform 은 oneM2M 표준을 준수하여 Http, MQTT, COAP, WebSocket 프로토콜과 Open API 를 제공한다. oneM2M 표준에 따라 Mobius 의 데이터 구조는 계층적인 리소스구조를 가지고 있으며, 기본적으로 REST API 방식을 제공합니다. Mobius Platform 은 HttpOpen API 를 위한 Http 서버와 MQTT 지원을 위한 MQTT 서버, COAP 을 위한 COAP 서버, WebSocket 지원을 위한 WebSocket 서버로 구성되어 있으며, 리소스 저장을 위한 DB 는 MySQL DBMS 를 사용한다.

1.2.2. Mobius 서버 플랫폼 주요 기능

Mobius Platform 은 디바이스와 어플리케이션을 연결하는 중간 매개체로서 디바이스는 서버 플랫폼으로 데이터를 전송하고, 플랫폼은 데이터를 저장한다. 어플리케이션은 Open API 를 통해 플랫폼에 저장된 데이터를 조회하고 플랫폼으로 제어 요청을 전송한다. 플랫폼은 설정된 디바이스에 따라 어플리케이션의 제어요청을 다시 디바이스로 전달한다.

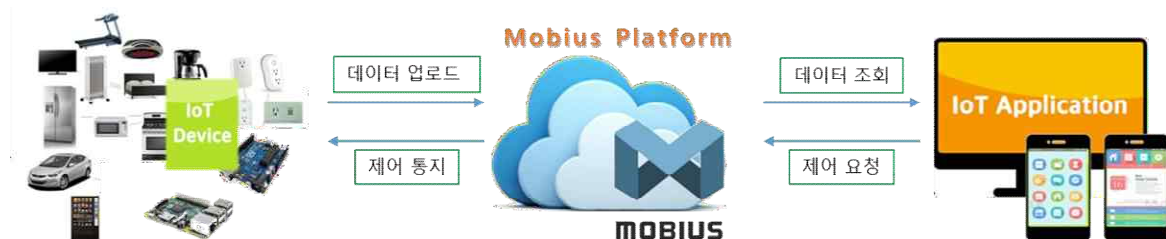


Figure 4 Mobius 서버 플랫폼 주요 기능

1.2.3. Mobius 서버 플랫폼 구성

Mobius Platform 은 디바이스와 어플리케이션을 연결하며, 디바이스와의 연결을 위해 HTTP, CoAP, MQTT, WebSocket 프로토콜을 지원하며, 리소스 저장을 위한 DB 는 MySQL 을 사용하여 구성된다

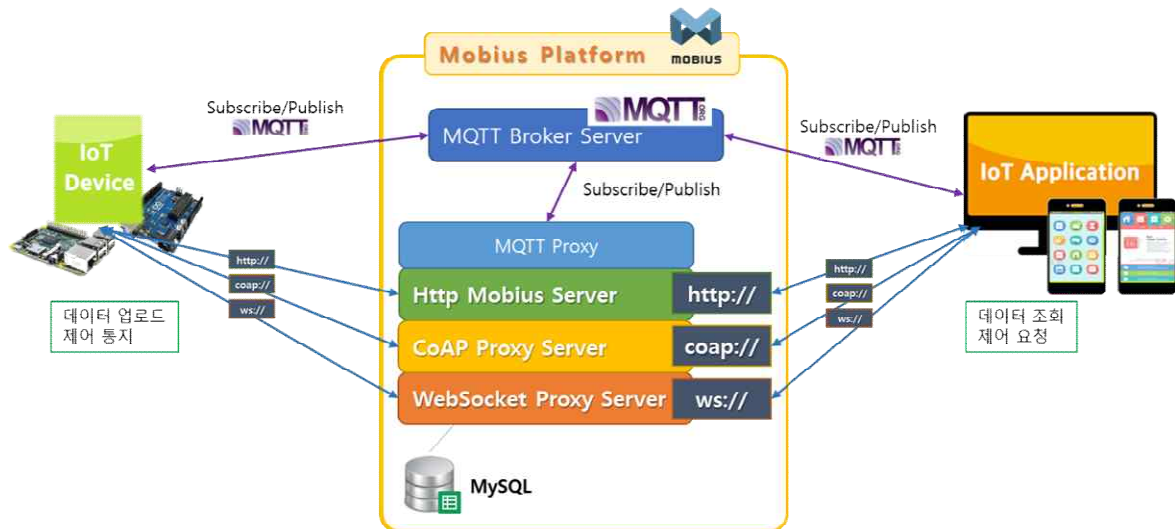


Figure 5 Mobius 서버 플랫폼 시스템 구성

1.2.4. Mobius 서버 플랫폼 S/W 아키텍처

Mobius 는 Node JS 를 기반으로 JavaScript 언어로 개발되었다. DB 는 MySQL 을 지원하고 있으며 통신 프로토콜은 HTTP, MQTT, CoAP, WebSocket 을 지원한다. 아래는 Mobius 의 소프트웨어 아키텍처를 나타낸다.

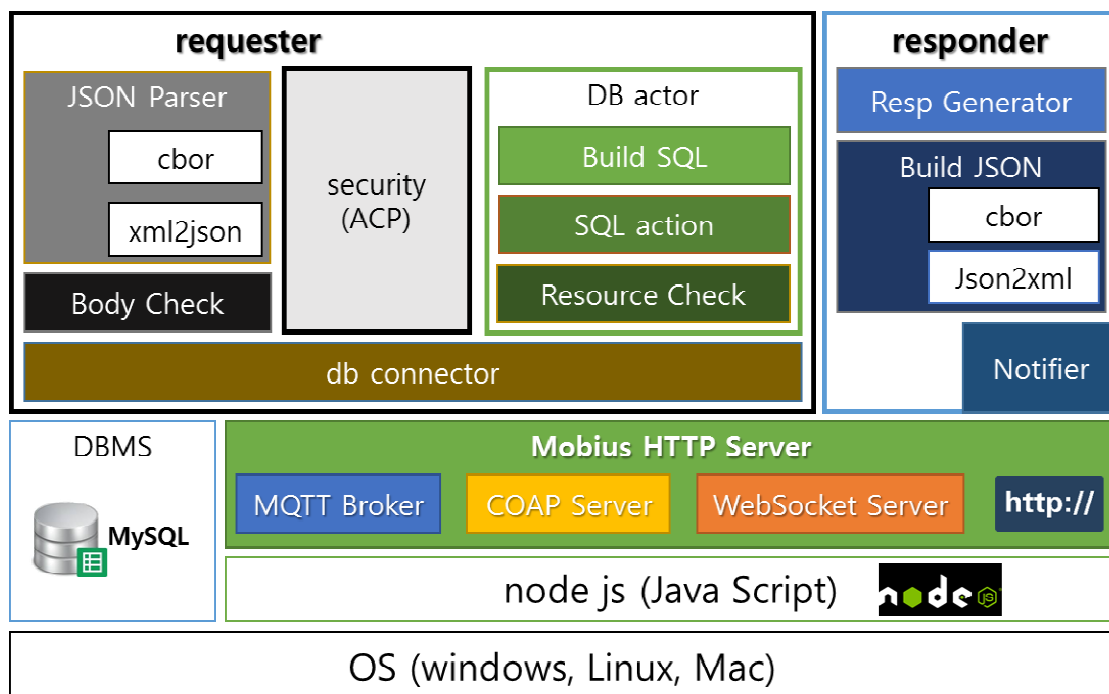


Figure 6 Mobius S/W 아키텍처

위의 Mobius SW 아키텍처는 컴포넌트 개념의 기능 아키텍처를 나타낸다. Mobius 플랫폼은 기본적으로 http 서버를 중심으로 MQTT Broker, COAP Server, WebSocket Server 가 구축된다. 위의 그림처럼 MQTT 의 경우 내부적으로 MQTT, HTTP 로 변환하는 proxy 를 구현하여 MQTT 를 지원하고 있고 COAP 과 WebSocket 도 같은 구조로 구현되어 있다.

Mobius 는 requester 와 responder 크게 두부분으로 구성되고 requester 는 DB 액세스를 위한 컴포넌트를 포함하고 있다. 모든 HTTP request 요청은 requestor 컴포넌트로 전달되고 parser-actor 를 거쳐 실제 데이터베이스에 데이터를 액세스하는 sql 문을 만들고 db connector 를 통해 데이터를 생성하거나 조회, 검색을 수행한다. DB 데이터 액세스 결과가 나오면 request 에 대한 응답으로 xml, json, cbor 의 데이터 포맷을 결정하고 responder 에 의해 처리된다.

Mobius 는 윈도우 파일 탐색기와 같은 트리 형태의 리소스 체계를 가지고 이를 지원하고 있다. 리소스 체계는 파일 탐색기의 폴더를 접근하는 것처럼 각 Resource 는 Resource 를 접근할 수 있는 URI 와 그대로 매핑된다. 예를 들어 IN-CSE 밑에, AE 밑에, container 밑의 데이터를 접근하는 URL 은 아래와 같다.

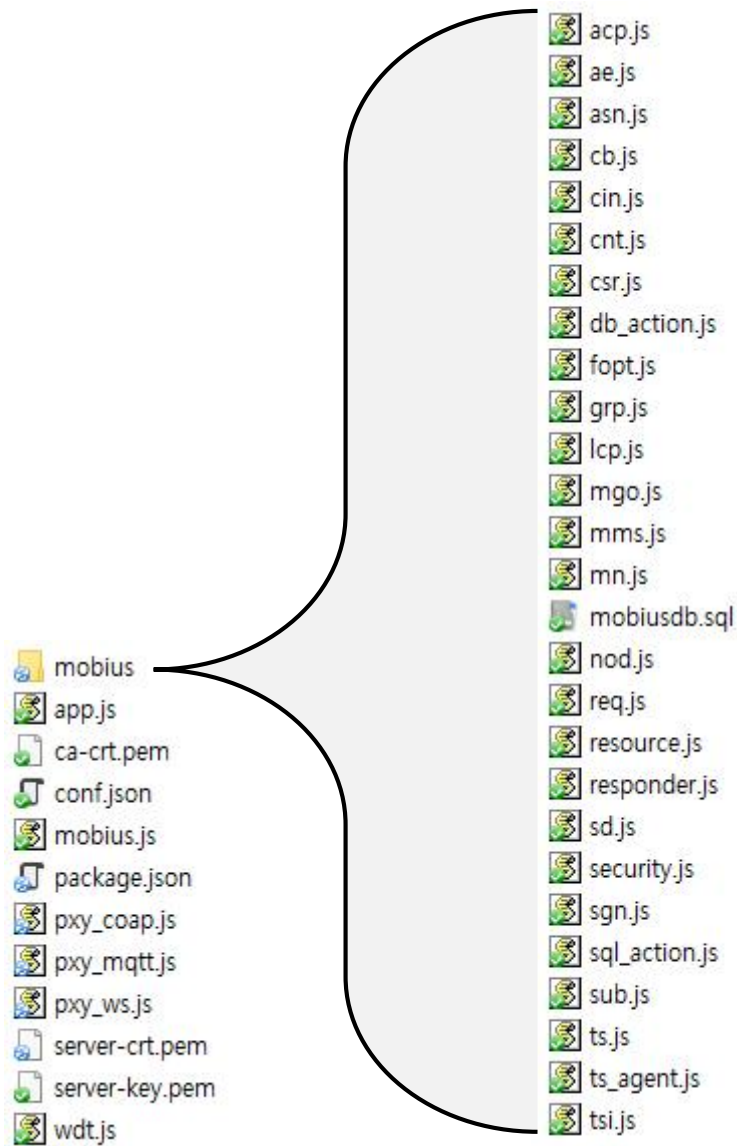
http://hostname:port/IN-CSE_name/AE_name/container_name/contentInstance_name

이처럼 모든 리소스는 위와 같은 방법으로 접근할 수 있으며 POST, GET, PUT, DELETE http method 를 통해 생성, 조회, 업데이트, 삭제 동작을 수행할 수 있다. (OCEAN 의 API문서 참조)

Container 밑에 container 를 생성할 수 있고 현재 URL 의 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 files 로 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	mobius 를 실행하는 파일로써 Mobius 구동을 위한 설정 값들을 수정할 수 있다. 이 파일을 통해 Mobius 의 메인 코드를 로드한다.
app.js	Mobius 의 main 역할을 수행하는 파일로 Mobius 로 들어오는 모든 패킷은 이 파일에서 제일 처음 처리한다. 이 파일에서 http 서버를 실행하고 Listen 하고 대기 한다. 이후 패킷이 들어오면 패킷의 타겟 url 을 파싱하고 패킷의 body 를 분석하여 정상적인 패킷인지 아닌지를 판단하고, 정상적인 패킷으로 판단되면 해당 동작을 수행하는 resource.js 로 전달한다. 오류 패킷일 경우 responder.js 를 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 topic 을 만들고 subscribe 한다. 이후 mqtt 로 데이터가 수신되면 이를 http 패킷으로 만들어서 Mobius 로 전달하고 response 을 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 Mobius 로 전달하고 response 을 받으면 이를 다시 coap 패킷으로 만들어 coap 으로 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 Mobius 로 전달하고 Mobius 로부터 response 을 받으면 이를 다시 websocket 패킷으로 만들어 websocket 으로 response 한다.
wdt.js	모비우스 자체 watchdogtimer 로 구축한 것이고, 모비우스 자체 타이머로 사용된다.
mobius/resource.js	이 파일은 mobius 가 지원하는 모든 리소스에 대해 create, retrieve, update, delete, notify, discovery 를 수행하는 코어 파일이다. app.js 에서 검증된 url, body 에 대해 해당되는 method 에 따라 내부적으로 처리하는 포맷으로 변환하고 실제 데이터베이스 연동 작업을 수행한다. 데이터 베이스 동작은 db_action.js 와 sql_action.js 를 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 얻기 위한 sql 문을 만들고 이를 db_action.js 의 함수를 통해 데이터베이스로부터 데이터를 리턴받아 전달하는 역할을 수행한다.
mobius/sgn.js	이 파일은 타겟 리소스에 subscription 이 차일로 존재하는지 체크하고 subscription 이 존재하면 해당 이벤트를 체크하고 subscription 의 notificationUri attribute 정보를 바탕으로 request 메시지를 만들어 전송하는 (notification 전송) 역할을 수행한다.
mobius/security.js	이 파일은 타겟 리소스에 대한 accessControlPolicy 를 체크하여 request 요청에 대해 reject 를 할 것인지 허용할 것인지를 판단한다. request 패킷의 X-M2M-Origin 헤더에 포함되어 있는 id 값을 가지고 타겟 리소스에 연결되어 있는 accessControlPolicy 리소스를 확인하여 이 id 에 접근 권한이 있는지 확인한다.
mobius/fopt.js	group 리소스에 대한 fanOutPoint 리소스에 대한 처리를 담당한다. fanOutPoint 리소스는 virtual 리소스로서 group 리소스에 포함된 멤버들 모두에게 요청받은 패킷을 전달하고 그 결과를 모아서 요청자에게 response 한다.
mobius/ts_agent.js	이 파일은 timeSeriesInstance 리소스를 관리하는 역할로서 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 Platform 은 MySQL DBMS 를 사용한다. 먼저 MySQL 을 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 (MacOS)

2.2.1. MySQL 설치

Mobius 는 데이터 저장 및 관리를 위해 MySQL DB 를 기반으로 동작한다. 여기서는 오픈 소스 RDBMS 인 MySQL 을 다운로드 받고 설치하는 방법에 대해 설명한다. MacOS 버전은 ARM 기반 Apple Silicon 프로세서를 사용하고 Ventura 13.2 버전으로 진행한다.

<http://dev.mysql.com/downloads/mysql>

MySQL 에 들어가 보면 MySQL 의 다양한 버전이 있는데 위 링크를 클릭해서 들어가면 무료로 사용할 수 있는 GPL 버전인 MySQL Community Server 를 다운받을 수 있다.

General Availability (GA) Releases Archives

MySQL Community Server 8.0.32

Select Operating System: **macOS** [Looking for previous GA versions?](#)

Select OS Version: **All**

ⓘ Packages for Ventura (13) are compatible with Monterey (12)

macOS 13 (ARM, 64-bit), DMG Archive (mysql-8.0.32-macos13-arm64.dmg)	8.0.32	553.4M	Download
macOS 13 (x86, 64-bit), DMG Archive (mysql-8.0.32-macos13-x86_64.dmg)	8.0.32	558.8M	Download
macOS 13 (ARM, 64-bit), Compressed TAR Archive (mysql-8.0.32-macos13-arm64.tar.gz)	8.0.32	174.7M	Download
macOS 13 (x86, 64-bit), Compressed TAR Archive (mysql-8.0.32-macos13-x86_64.tar.gz)	8.0.32	179.2M	Download
macOS 13 (ARM, 64-bit), Compressed TAR Archive Test Suite (mysql-test-8.0.32-macos13-arm64.tar.gz)	8.0.32	378.5M	Download
macOS 13 (x86, 64-bit), Compressed TAR Archive Test Suite (mysql-test-8.0.32-macos13-x86_64.tar.gz)	8.0.32	379.1M	Download
macOS 13 (ARM, 64-bit), TAR (mysql-8.0.32-macos13-arm64.tar)	8.0.32	570.0M	Download
macOS 13 (x86, 64-bit), TAR (mysql-8.0.32-macos13-x86_64.tar)	8.0.32	576.2M	Download

ⓘ We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.

여기서는 MacOS 에 MySQL 을 설치하기 위해 macOS 13(Apple Silicon-ARM 버전, Intel 프로세서-x86 버전)이라고 적힌 Download 버튼을 클릭한다. (MySQL 에는 MySQL Enterprise Edition 이나 MySQL Cluster CGE 처럼 유료로 사용해야 하는(commercial) 것도 있으니 참고하길 바란다.)

MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

Login »

using my Oracle Web account

Sign Up »

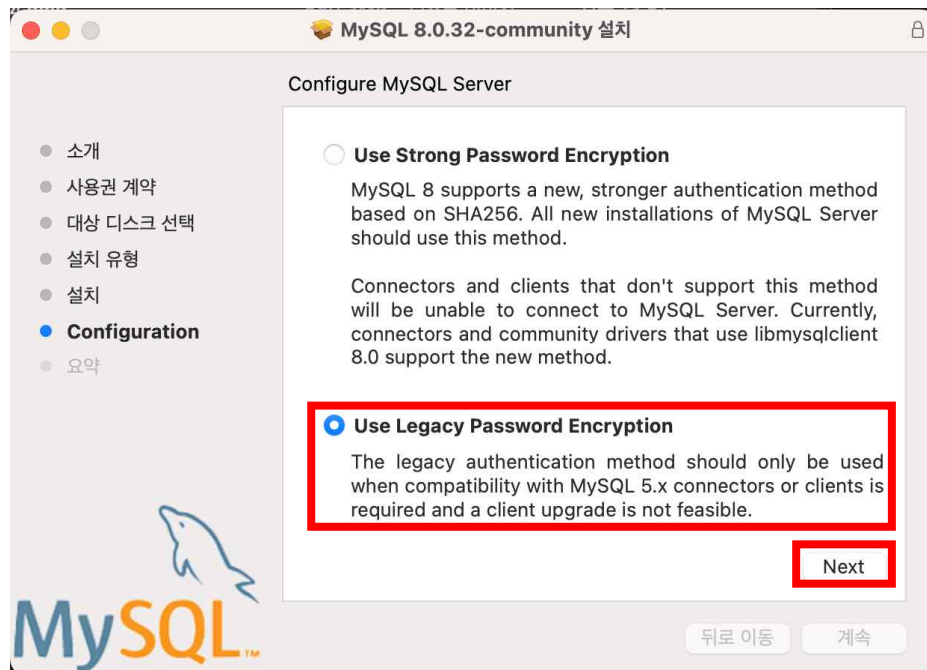
for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can sign up for a free account by clicking the Sign Up link and following the instructions.

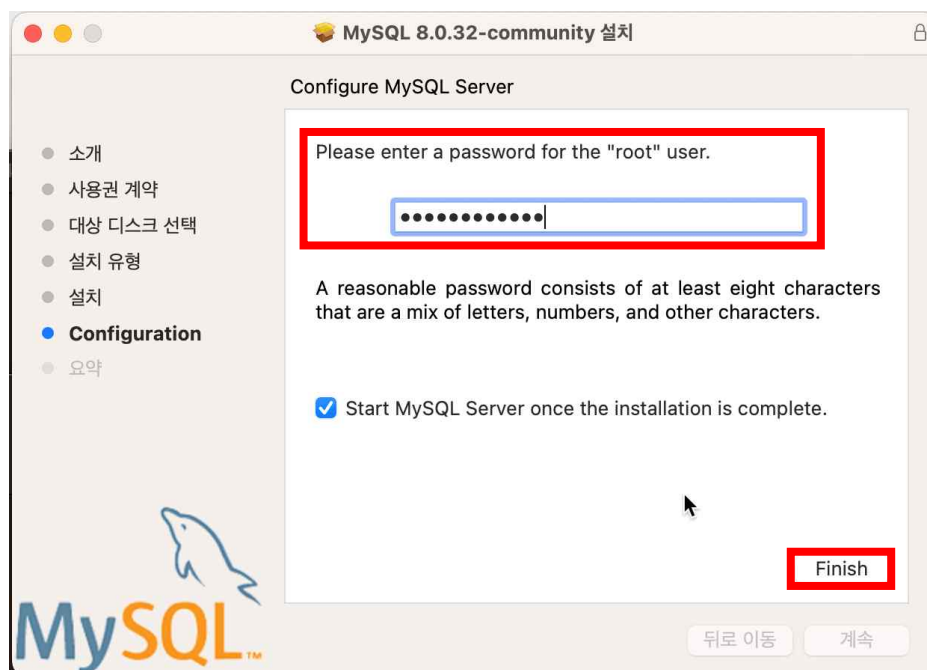
No thanks, just start my download.

로그인 화면이 나오면 회원가입을 하거나 아래에 있는 **No thanks, just start my download** 링크를 클릭해서 다운로드 받는다.

다운받은 .dmg 확장명의 MySQL 디스크 이미지 파일을 실행시켜 .pkg 확장명의 설치파일을 실행한 후 라이선스에 동의하고 설치 버튼을 누른다.

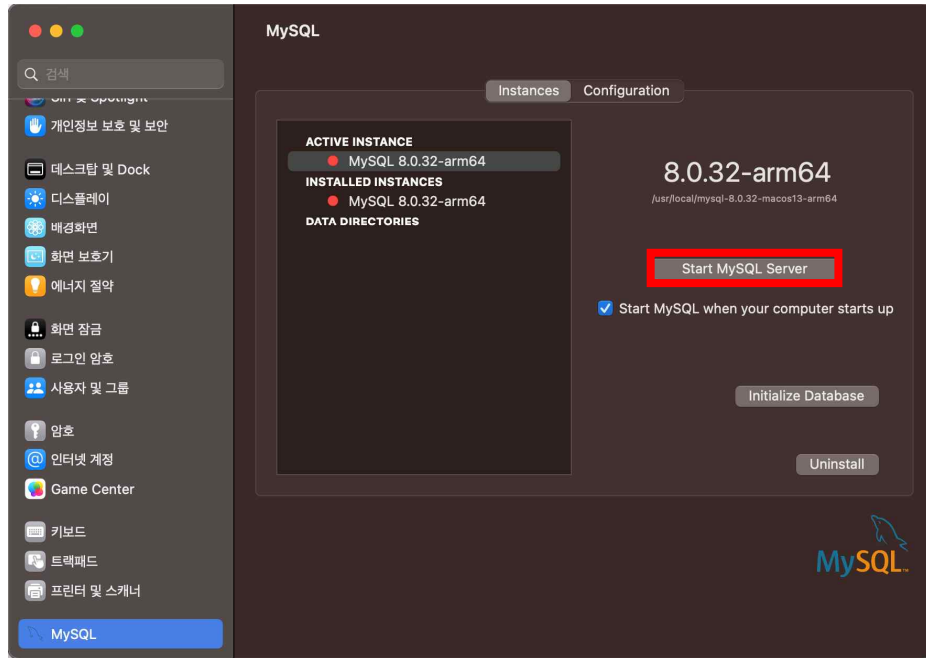


MySQL 8.0 부터 새로운 사용자 인증 방식이 적용되었는데 Mobius 연동 및 기존 프로그램들과 호환을 위해 꼭 **Use Legacy Password Encryption** 으로 설치를 진행하여야 한다.

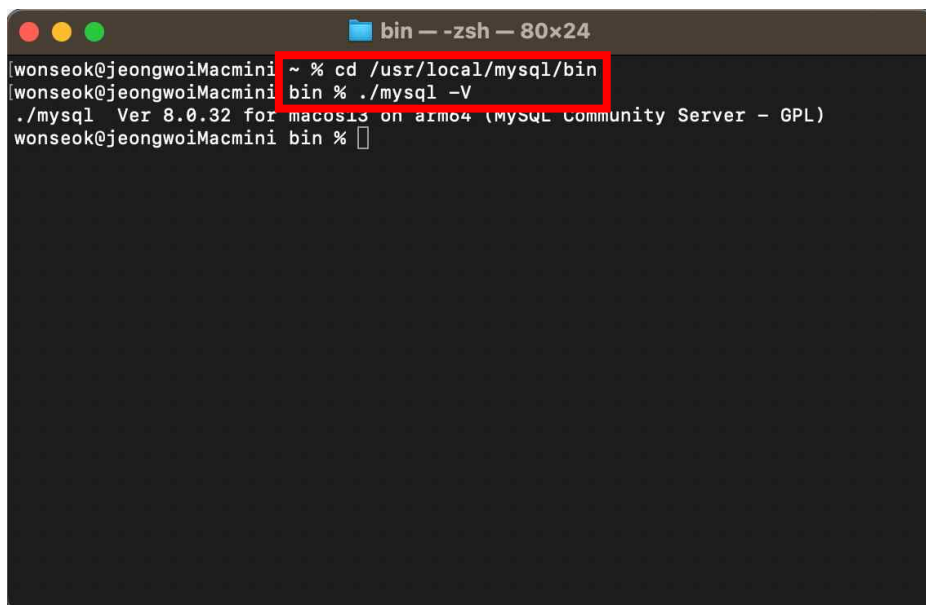


루트암호(Root Password)를 지정한 후 Finish 를 클릭한다.

Mobius 와 MySQL 을 연동할 때 필요하므로 꼭 기억하여야 한다.



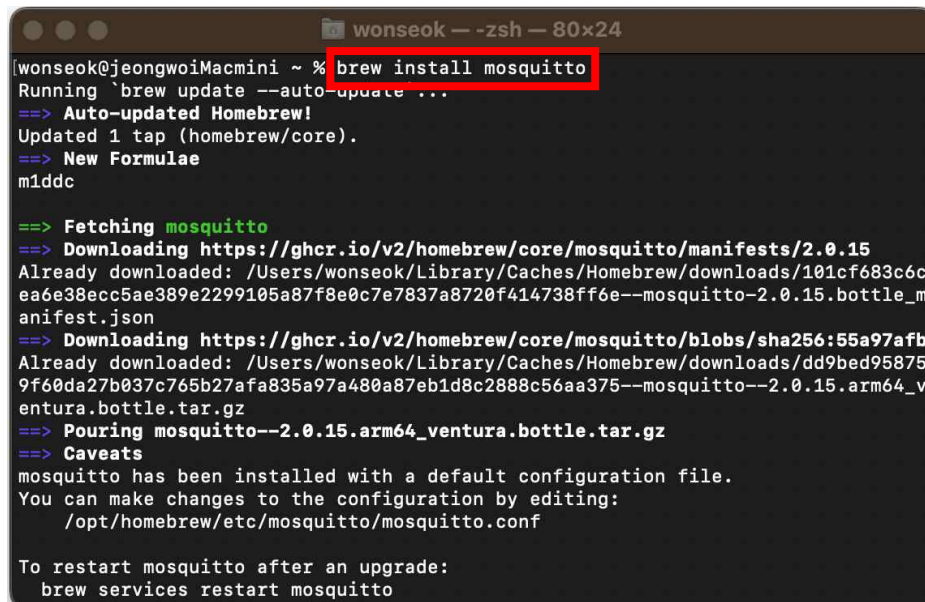
시스템 설정에서 MySQL 항목을 찾아 위의 버튼을 클릭하여 동작시킨다.



위의 명령어와 같이 입력하여 MySQL 이 설치된 경로로 이동 후 “./mysql -V” 명령어로 버전을 확인하여 설치를 확인한다.

2.2.1. MQTT 브로커 설치

MQTT(Message Queuing Telemetry Transport)는 발행-구독 기반의 메시징 프로토콜이다. Mobius Platform 은 디바이스와의 통신을 위해 MQTT 프로토콜을 지원하고 있으며 이를 위해서 오픈소스로 제공되는 MQTT 브로커인 Mosquitto 서버를 설치한다.

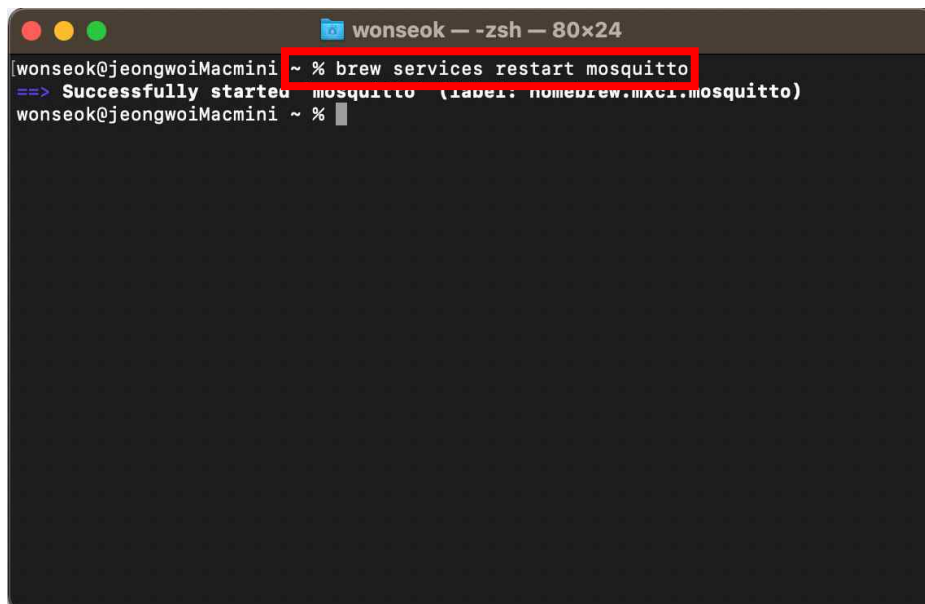


```
wonseok@jeongwoiMacmini ~ % brew install mosquitto
Running `brew update --auto-update ...
==> Auto-updated Homebrew!
Updated 1 tap (homebrew/core).
==> New Formulae
m1ddc

==> Fetching mosquitto
==> Downloading https://ghcr.io/v2/homebrew/core/mosquitto/manifests/2.0.15
Already downloaded: /Users/wonseok/Library/Caches/Homebrew/downloads/101cf683c6c
ea6e38ecc5ae389e2299105a87f8e0c7e7837a8720f414738ff6e--mosquitto-2.0.15.bottle_m
anifest.json
==> Downloading https://ghcr.io/v2/homebrew/core/mosquitto/blobs/sha256:55a97afb
9f60da27b037c765b27afa835a97a480a87eb1d8c2888c56aa375--mosquitto--2.0.15.arm64_v
entura.bottle.tar.gz
==> Pouring mosquitto--2.0.15.arm64_ventura.bottle.tar.gz
==> Caveats
mosquitto has been installed with a default configuration file.
You can make changes to the configuration by editing:
/opt/homebrew/etc/mosquitto/mosquitto.conf

To restart mosquitto after an upgrade:
brew services restart mosquitto
```

MacOS 에서 아래의 "brew install mosquitto" 명령어를 통해 mosquitto 설치가 가능하다.

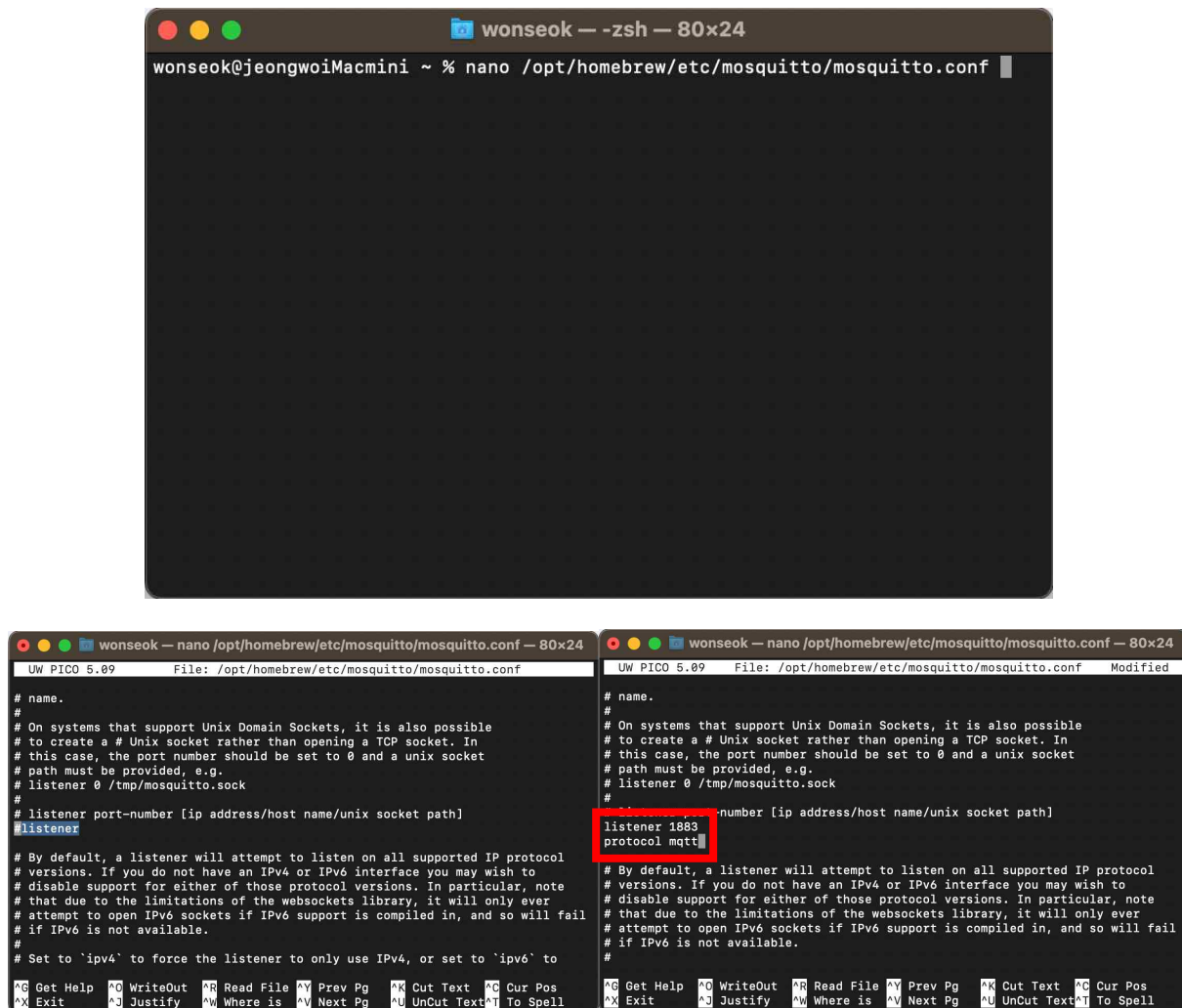


```
wonseok@jeongwoiMacmini ~ % brew services restart mosquitto
==> Successfully started mosquitto (label: homebrew.mxcl.mosquitto)
wonseok@jeongwoiMacmini ~ %
```

설치된 mosquitto 서비스 실행을 위해 아래와 같이 "brew services restart mosquitto" 명령어를 입력한다.

mosquitto 는 2.0 버전으로 업데이트 되면서 기존의 기본 설정을 사용하는 대신 사용자가 설정한 내용을 기반으로 동작한다. 이를 위해 mosquitto.conf 파일을 찾아 아래와 같이 설정한다.

mosquitto.conf 파일은 mosquitto 를 설치할 때 설정한 경로(/opt/homebrew/etc/mosquito/mosquito.conf)의 폴더 안에 있다. 설정 파일 수정을 위해 nano 에디터를 통해 아래와 같이 입력한다.



"#listener"를 검색하여 오른쪽 그림과 같이 "listener 1883", "protocol mqtt"으로 수정한다.

```

# Boolean value that determines whether clients that connect
# without providing a username are allowed to connect. If set to
# false then a password file should be created (see the
# password_file option) to control authenticated client access.
#
# Defaults to false, unless there are no listeners defined in the configuration
# file, in which case it is set to true, but connections are only allowed from
# the local machine.
allow_anonymous false

#
# Default authentication and topic access control
#
# Control access to the broker using a password file. This file can be
# generated using the mosquitto_passwd utility. If TLS support is not compiled
# into mosquitto (it is recommended that TLS support should be included) then
# plain text passwords are used, in which case the file should be a text file

```

“#allow_anonymous false”를 검색하여 오른쪽 그림과 같이 “allow_anonymous true”로 수정한다.

```

wonseok@jeongwoiMacmini ~ % nano /opt/homebrew/etc/mosquitto/mosquitto.conf
wonseok@jeongwoiMacmini ~ % brew services restart mosquitto
Stopping `mosquitto`... (might take a while)
==> Successfully stopped `mosquitto` (label: homebrew.mxcl.mosquitto)
==> Successfully started `mosquitto` (label: homebrew.mxcl.mosquitto)
wonseok@jeongwoiMacmini ~ %

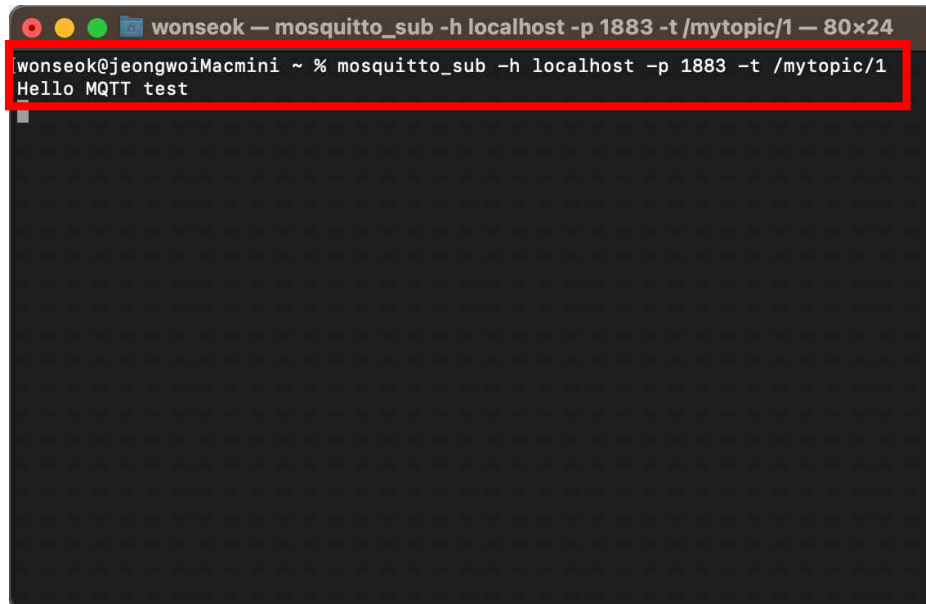
```

변경한 설정 파일을 저장 후 “brew services restart mosquitto” 명령어를 통해 mosquitto 를 재실행한다.

mosquitto 가 제대로 설치됐는지 테스트하기 위해 두 개의 터미널을 실행시키고 다음과 같이 명령어를 실행한다.

구독 (Subscribe)

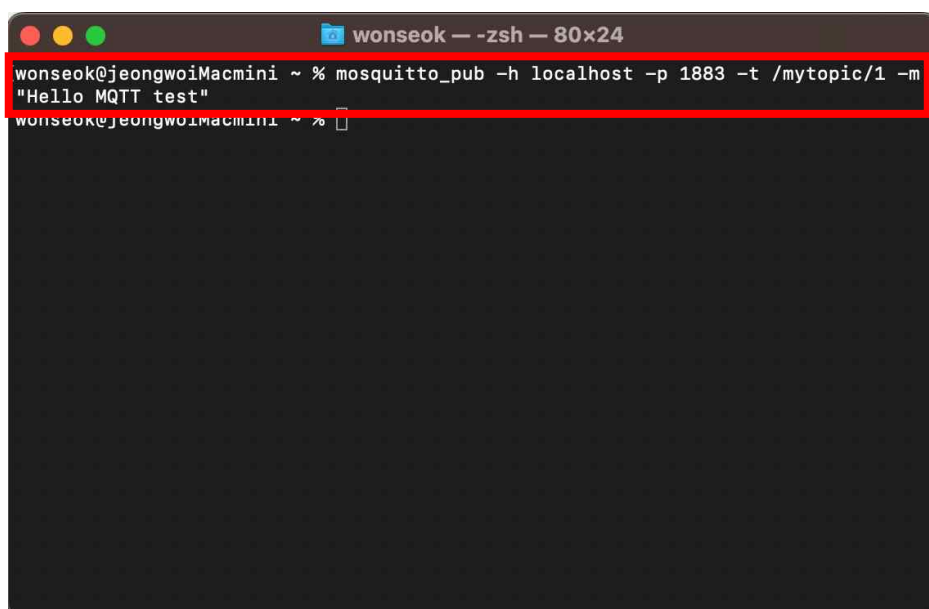
```
% mosquitto_sub -h localhost -t /mytopic/1
```



```
wonseok — mosquitto_sub -h localhost -p 1883 -t /mytopic/1 — 80x24
wonseok@jeongwoiMacmini ~ % mosquitto_sub -h localhost -p 1883 -t /mytopic/1
Hello MQTT test
```

발행 (Publish)

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

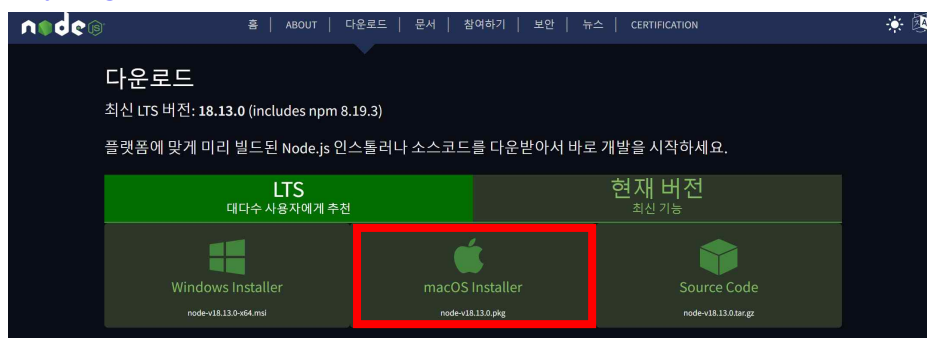


```
wonseok — zsh — 80x24
wonseok@jeongwoiMacmini ~ % mosquitto_pub -h localhost -p 1883 -t /mytopic/1 -m
"Hello MQTT test"
wonseok@jeongwoiMacmini ~ %
```

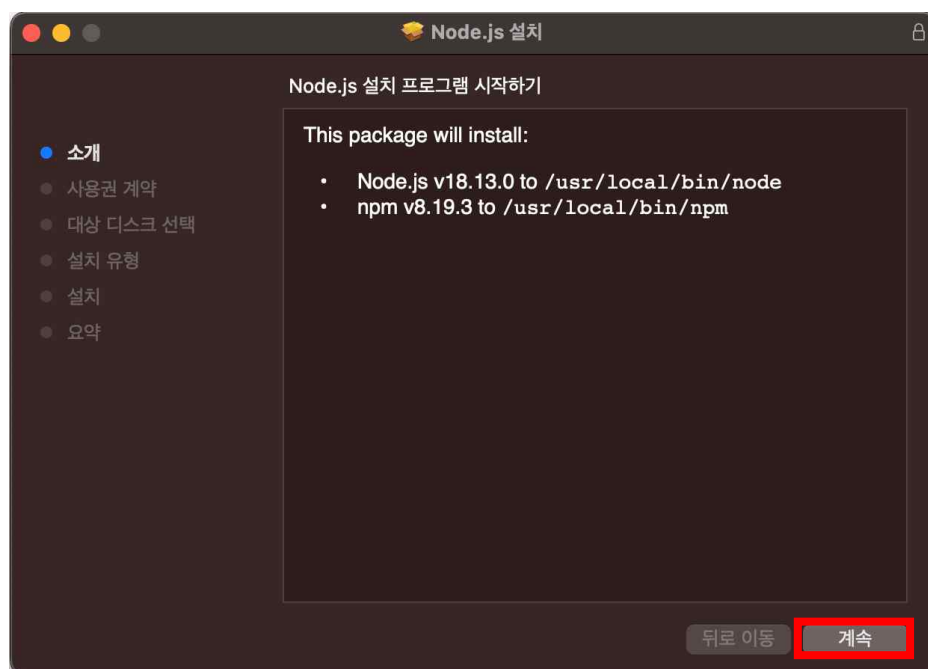

2.2.2. Node JS 설치

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

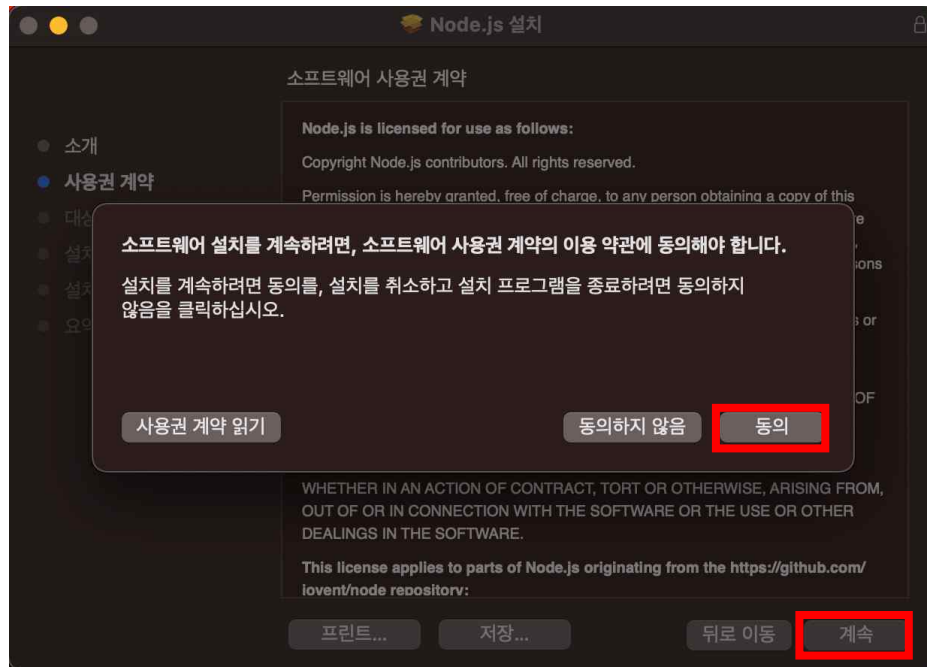
<https://nodejs.org/download/release/v16.19.0/>



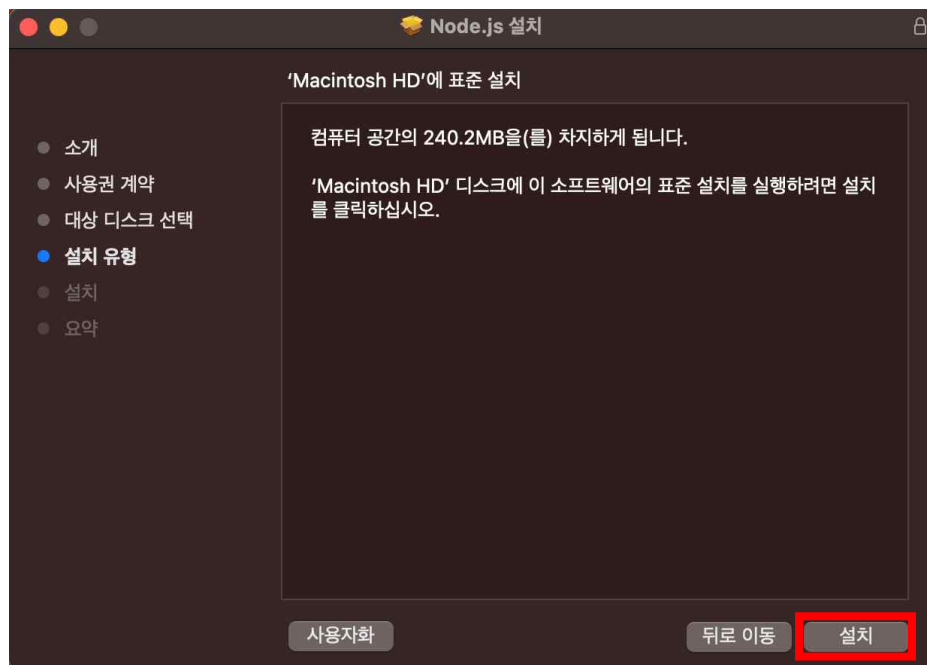
위 링크에서 최신버전의 node.js 설치파일을 다운받는다.



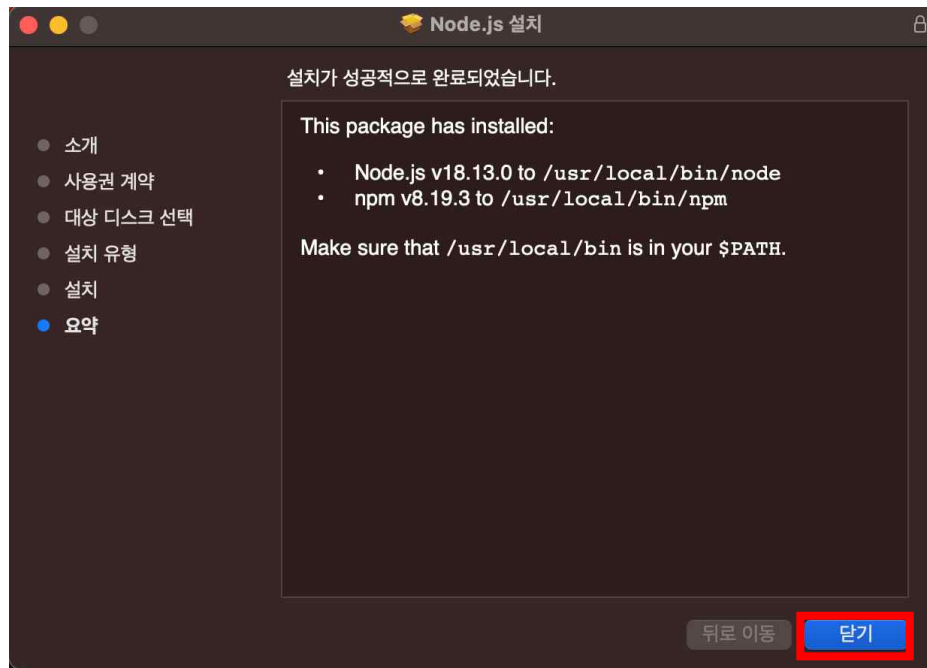
다음으로 위와 같이 다운로드한 .pkg 확장명의 설치파일을 실행한다. 계속을 클릭한다.



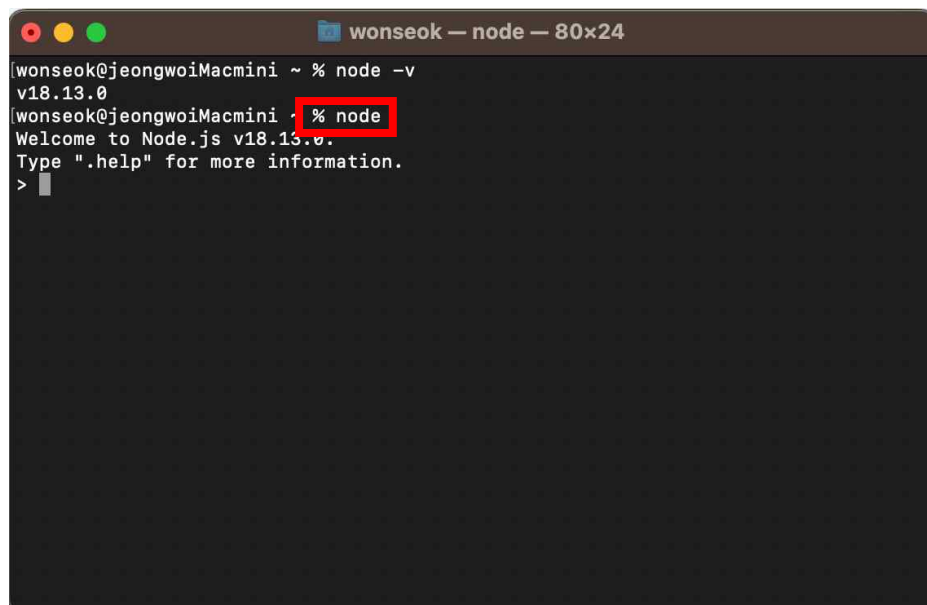
소프트웨어 사용권 계약 화면에서 계속을 클릭한 후 동의를 클릭하여 진행한다.



설치를 클릭한다.



설치가 완료되면 Finish 버튼을 클릭한다.



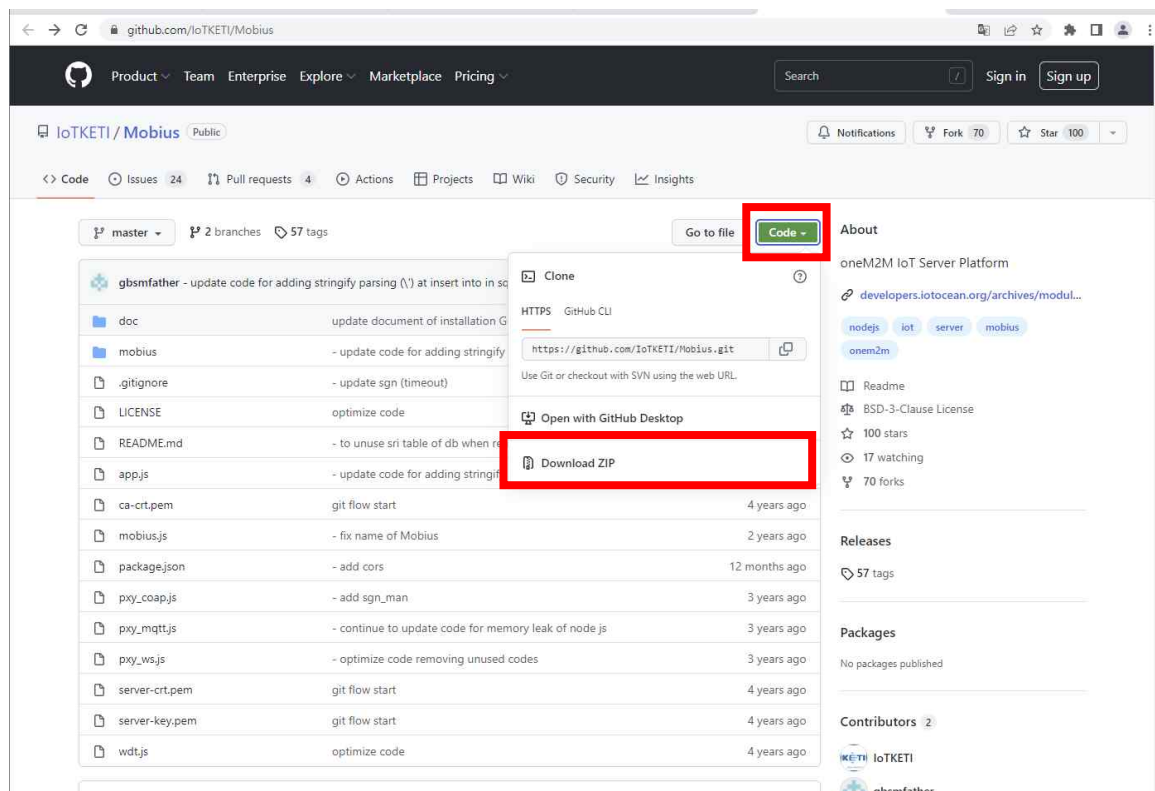
Node.js 가 제대로 설치가 되었는지 확인한다. Node.js 는 대화형 cli 를 제공한다. 위와 같이 실행이 되면 Node.js 가 제대로 설치가 된 것이다.

2.3. Mobius 설치

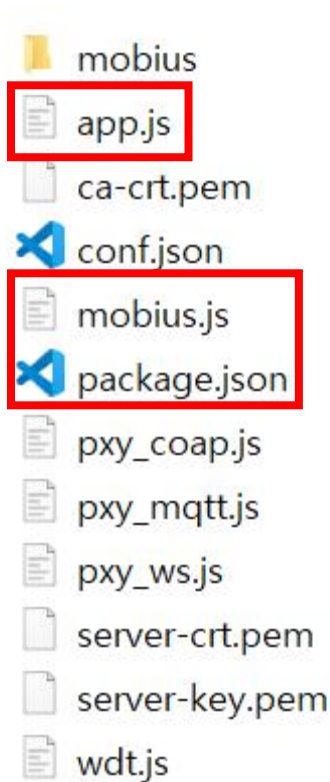
Mobius Platform은 GitHub를 통해 Mobius 서버 플랫폼 소스를 공개하고 있다. Mobius 서버 플랫폼 압축 파일을 다운로드 받은 후, 제공된 매뉴얼에 따라 Port 정보 등의 Configuration 정보를 설정한다.

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.



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 풀 후 실행할 수 있다. 이전의 MySQL과 Node.js가 제대로 설치가 되어 있다면 Mobius에 사용된 node.js 모듈을 설치하면 별도의 작업없이 실행이 가능하다.



mobius.js 가 Mobius 의 메인 코드를 실행하는 파일이다. app.js 는 대문역할과 라우팅 역할을 수행하는 메인 코드이다. mobius 폴더에 있는 파일들이 실제 패킷을 처리하고 DB 액세스와 응답을 수행하는 코드들이다.

현재는 추가적인 node.js 모듈들이 설치가 되어 있지 않아 에러가 날 것이다. 다음과 같이 명령 프롬프트를 실행시키고 npm 툴을 이용해 필요한 모듈을 설치해야 한다. package.json 파일에 필요한 모듈 리스트를 가지고 있다.

```

wonseok@jeongwoiMacmini Mobius % npm install
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.
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy.
new code should use the URLSearchParams API instead.

added 184 packages, and audited 185 packages in 1s

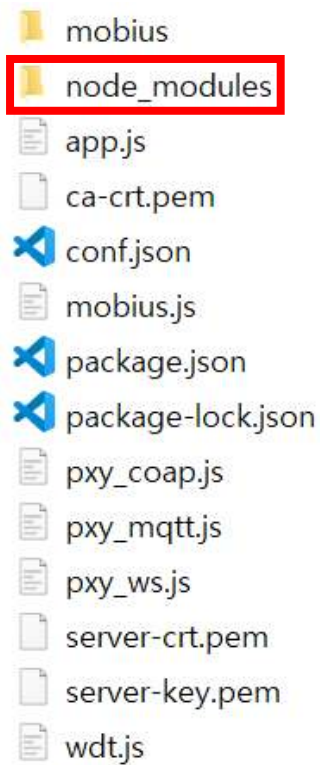
9 packages are looking for funding
  run `npm fund` for details

5 high severity vulnerabilities

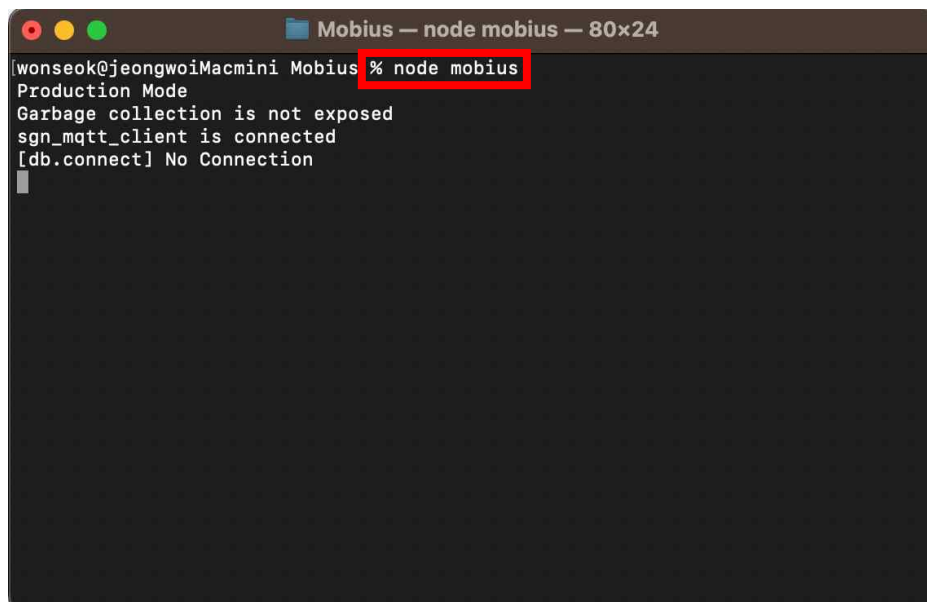
To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
npm notice
npm notice New major version of npm available! 8.19.3 -> 9.4.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v9.4.0
npm notice Run npm install -g npm@9.4.0 to update!
npm notice
wonseok@jeongwoiMacmini Mobius %
```

다운로드 받은 Mobius 폴더에서 위와 같이 npm install 명령어를 입력하면 필요한 모듈들이 설치가 된다.



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



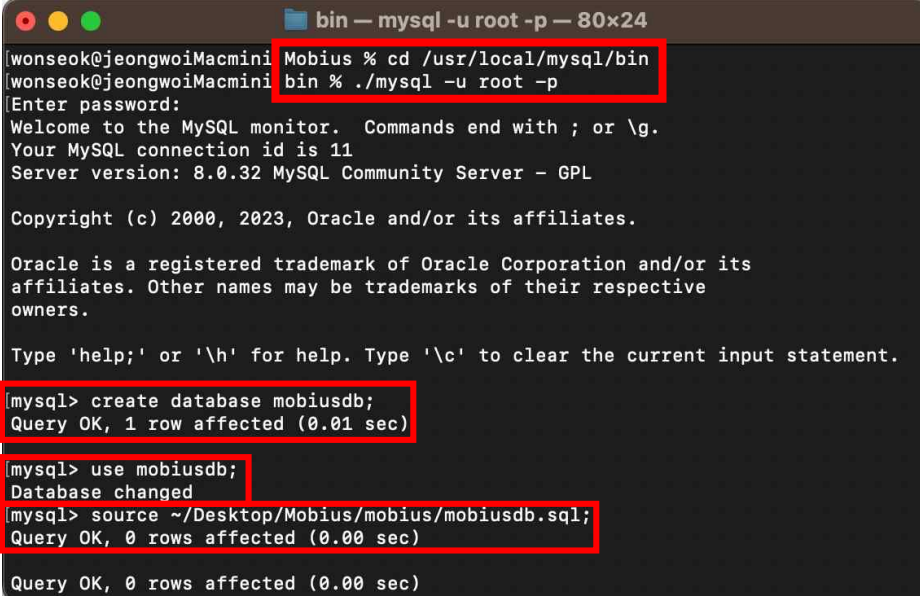
이제 Mobius 를 구동하기 위한 모듈 설치가 끝났다.

하지만 위의 그림과 같이 node mobius.js 를 실행하면 MySQL 과 데이터베이스가 연동되어
있지 않아 오류가 발생한다.

따라서 Mobius 와 MySQL 을 연동하는 과정이 필요하다.

2.3.2. mobiusdb 데이터베이스 생성(MacOS)

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



```
bin — mysql -u root -p — 80x24
wonseok@jeongwoiMacmini Mobius % cd /usr/local/mysql/bin
wonseok@jeongwoiMacmini 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: 8.0.32 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.

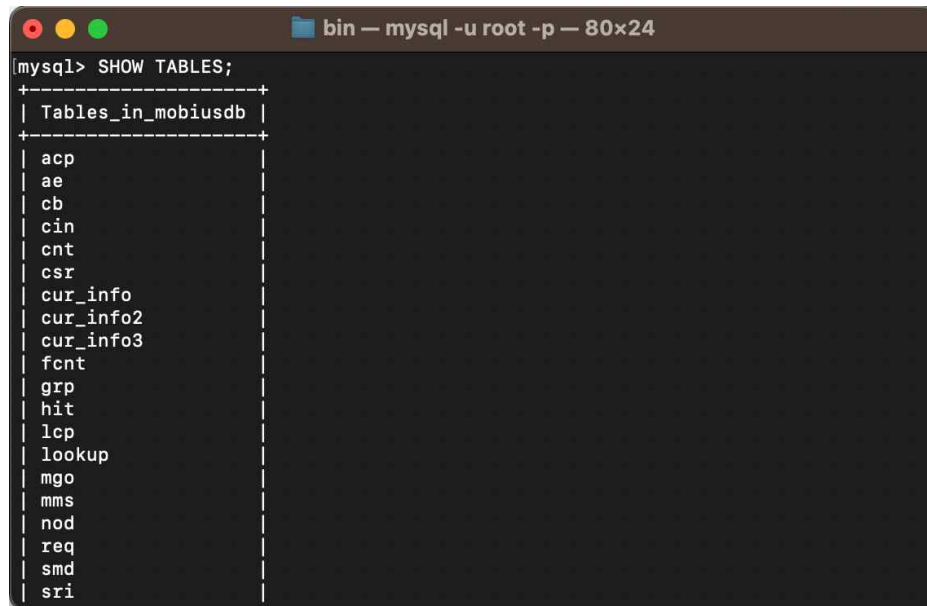
mysql> create database mobiusdb;
Query OK, 1 row affected (0.01 sec)

mysql> use mobiusdb;
Database changed
mysql> source ~/Desktop/Mobius/mobius/mobiusdb.sql;
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

먼저 "mysql -u root -p" 명령어를 입력하여 root 계정으로 로그인한다. 이후 패스워드를 입력하라는 메시지가 나타나면 MySQL 을 설치할 때 생성했던 패스워드를 입력한다. Mobius 와 연동을 위한 MySQL 설정은 다음과 같은 순서로 SQL 문을 입력하여 진행한다.

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



```
bin — mysql -u root -p — 80x24
mysql> SHOW TABLES;
+-----+
| Tables_in_mobiusdb |
+-----+
| acp                 |
| ae                  |
| cb                  |
| cin                 |
| cnt                 |
| csr                 |
| cur_info             |
| cur_info2           |
| cur_info3           |
| fcmt                |
| grp                 |
| hit                 |
| lcp                 |
| lookup              |
| mgo                 |
| mms                 |
| nod                 |
| req                 |
| smd                 |
| sri                 |
+-----+
```

MySQL 에 정상적으로 테이블이 생성되었는지 확인을 위해 "SHOW TABLES;" 명령어를 입력한다.

3. Mobius 서버 플랫폼 구동

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

3.1. 환경설정

Mobius 의 환경 설정은 conf.json 파일을 통해 설정을 변경할 수 있다.

```
{  
  "csebaseport": "7579",  
  "dbpass": "dksd1fduq2"  
}
```

Mobius 구동 시 필요한 설정은 위와 같이 Mobius 가 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.

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


```

wonseok@jeongwoiMacmini Mobiu % node mobius.js
Production Mode
Garbage collection is not exposed
sgn_mqtt_client is connected
[CPU Count: 8]
select_ri_lookup /Mobius: 1.695ms
insert_cb /Mobius: 3.159ms
""
Production Mode
Garbage collection is not exposed
Production Mode
Garbage collection is not exposed
Garbage collection is not exposed
Production Mode
Garbage collection is not exposed
Production Mode
Garbage collection is not exposed
sgn_mqtt_client is connected
sgn_mqtt_client is connected
sgn_mqtt_client is connected
sgn_mqtt_client is connected
mobius server (192.168.50.114) running at 7579 port
mobius server (192.168.50.114) running at 7579 port
select_ri_lookup /Mobius: 1.329ms
select_ri_lookup /Mobius: 0.975ms
update_cb_poa_csi /Mobius: 0.528ms
""
update_cb_poa_csi /Mobius: 0.734ms
""
mobius server (192.168.50.114) running at 7579 port
sgn_mqtt_client is connected
mobius server (192.168.50.114) running at 7579 port
mobius server (192.168.50.114) running at 7579 port
select_ri_lookup /Mobius: 3.343ms
update_cb_poa_csi /Mobius: 3.457ms
""
select_ri_lookup /Mobius: 1.063ms
select_ri_lookup /Mobius: 4.952ms
update_cb_poa_csi /Mobius: 0.816ms
""
update_cb_poa_csi /Mobius: 0.799ms
""
Production Mode
Garbage collection is not exposed
Production Mode
Garbage collection is not exposed
sgn_mqtt_client is connected
sgn_mqtt_client is connected
mobius server (192.168.50.114) running at 7579 port
Production Mode
Garbage collection is not exposed
mobius server (192.168.50.114) running at 7579 port
select_ri_lookup /Mobius: 0.778ms
update_cb_poa_csi /Mobius: 0.313ms
sgn_mqtt_client is connected
""
select_ri_lookup /Mobius: 0.763ms
update_cb_poa_csi /Mobius: 0.264ms
""
mobius server (192.168.50.114) running at 7579 port
select_ri_lookup /Mobius: 0.63ms
update_cb_poa_csi /Mobius: 0.285ms
""
pxyws server (192.168.50.114) running at 7577 port

GET : /Mobius

GET : /Mobius
acpi
acpi
lbl
lbl
at
at
aa
subl
subl
srt
srt
poa
poa
srv
srv
get_resource_from_url (0i7Bb-tkX) - /Mobius: 4.671ms
get_resource_from_url (4i0RdBOQ) - /Mobius: 4.665ms
{"m2m:cb":{"pi":null,"ri":"5-20230126112534668660","ty":5,"ct":"20230126T112534","rn":"Mobius","lt":"20230126T112534","lb1":["Mobius"],"cst":1,"csi":["Mobius2"],"srt":[1,2,3,4,5,9,10,13,14,16,17,23],"poa":["http://192.168.50.114:7579"],"sr v":{"1","2","2a"}}}
{"m2m:cb":{"pi":null,"ri":"5-20230126112534668660","ty":5,"ct":"20230126T112534","rn":"Mobius","lt":"20230126T112534","

```

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

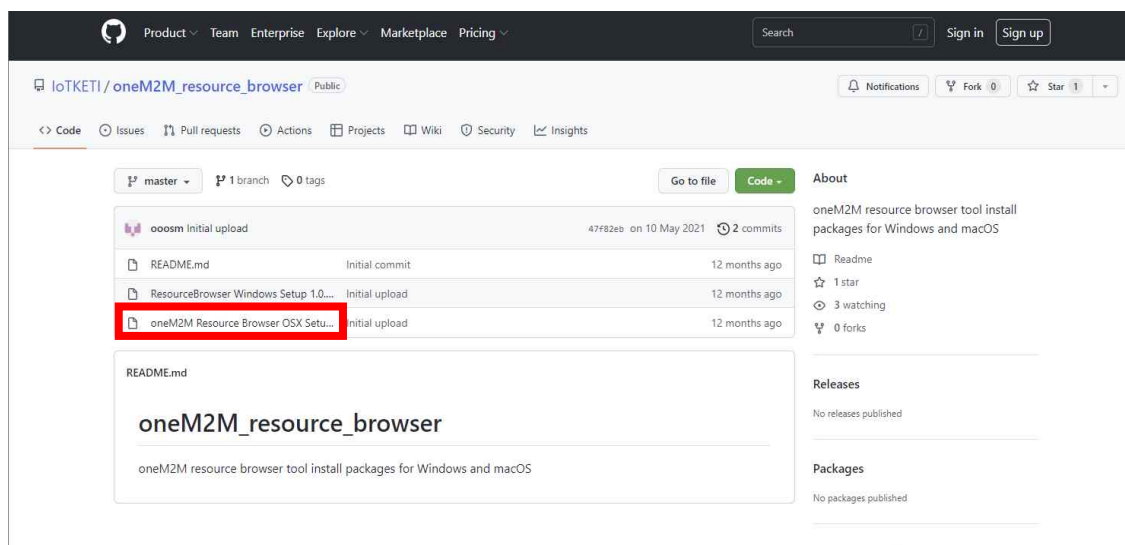
3.2.1. 리소스뷰어(oneM2M Browser) 활용

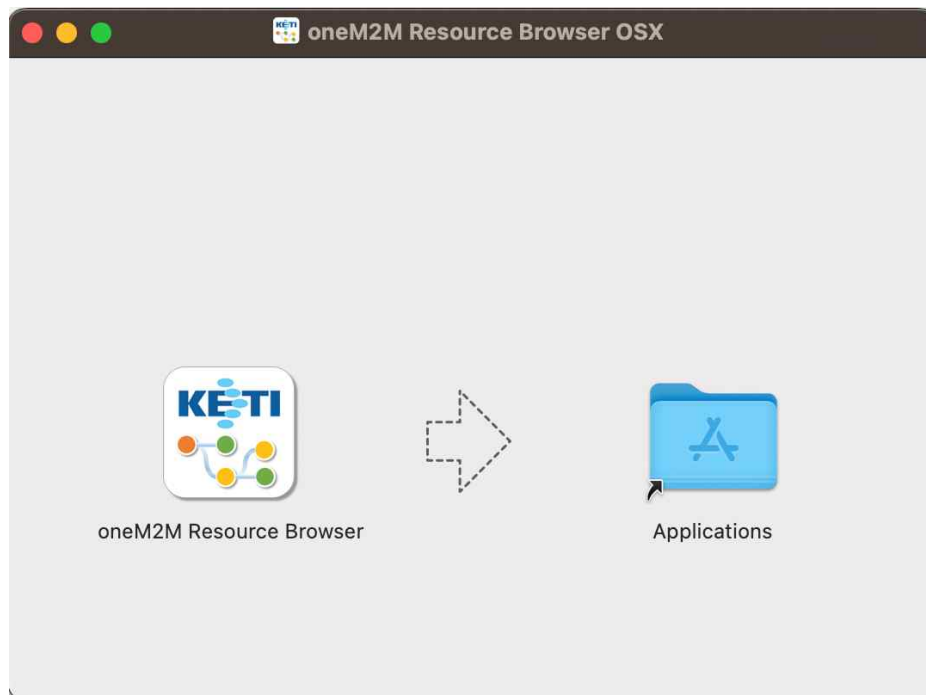
Mobius 와 nCube 를 통해 만들어진 디바이스가 제대로 동작하는지 테스트하기 위해 웹으로 제공하는 리소스뷰어를 사용할 수 있다. 리소스뷰어는 윈도우 파일 탐색기와 비슷하게 트리 구조의 리소스를 보여주고 리소스 생성, 삭제의 간단한 동작을 수행할 수 있다.

웹버전의 리소스뷰어는 아래의 url 로 제공하고 있다.

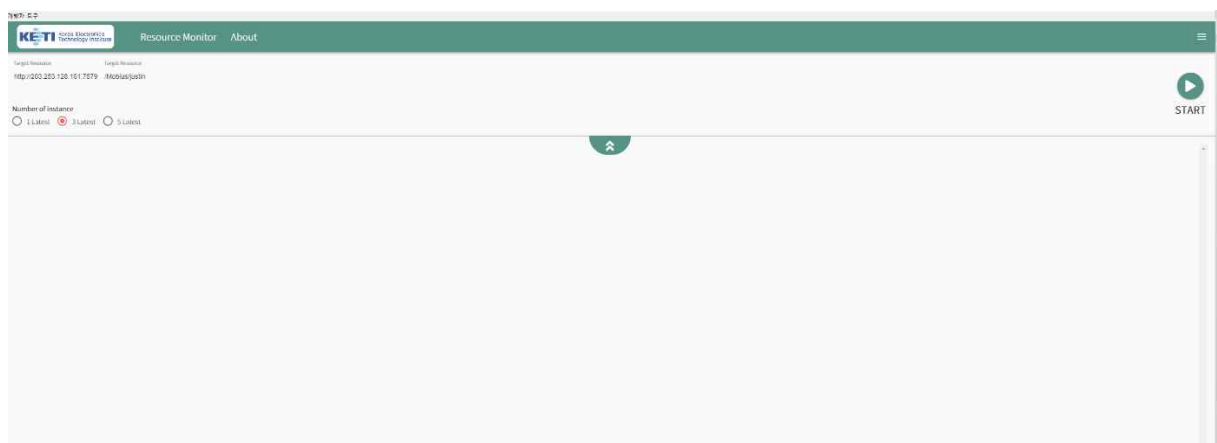
<http://203.253.128.161:7575>

웹버전의 리소스뷰어 Mobius 의 리소스만을 볼 수 있는 도구이고 다른 oneM2M 플랫폼의 구조를 볼 수 있는 버전은 OCEAN 의 GitHub 에 oneM2M Browser 로 제공하고 있다.

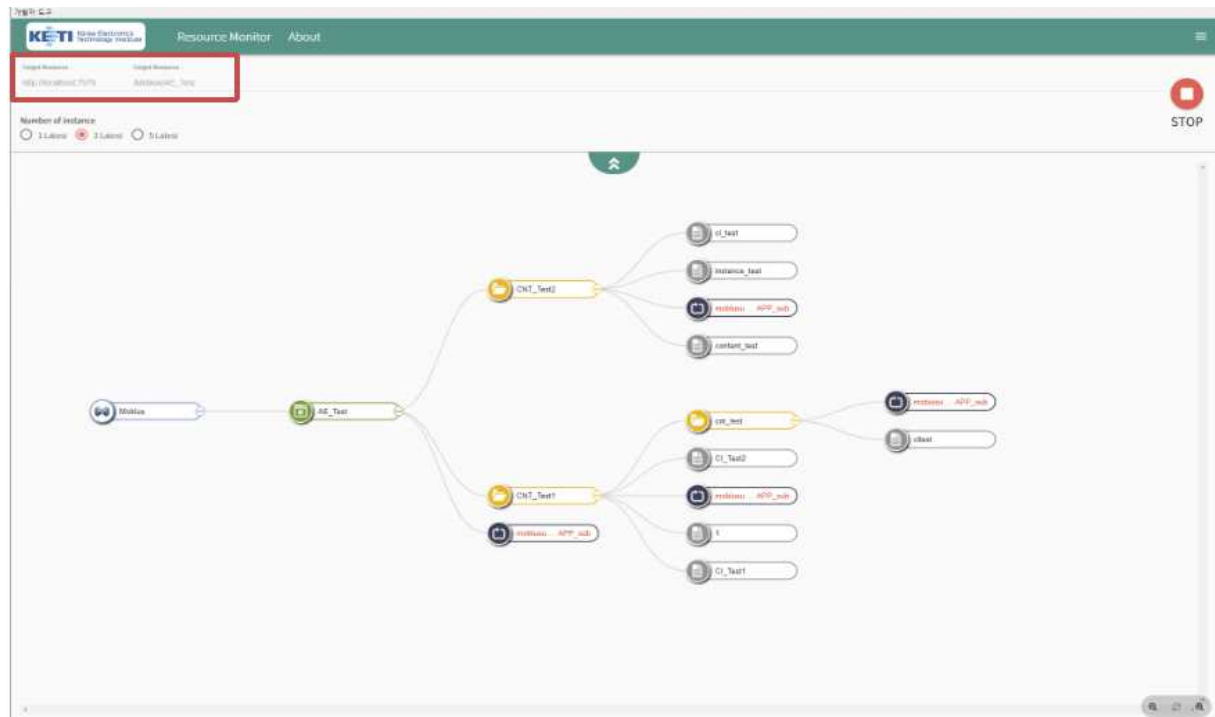




GitHub 에서 다운로드 받은 .dmg 확장명의 디스크 이미지 파일을 클릭한 후, 위와 같은 화면에서 oneM2M Resource Browser 아이콘을 Applications 로 드래그하여 설치한다.



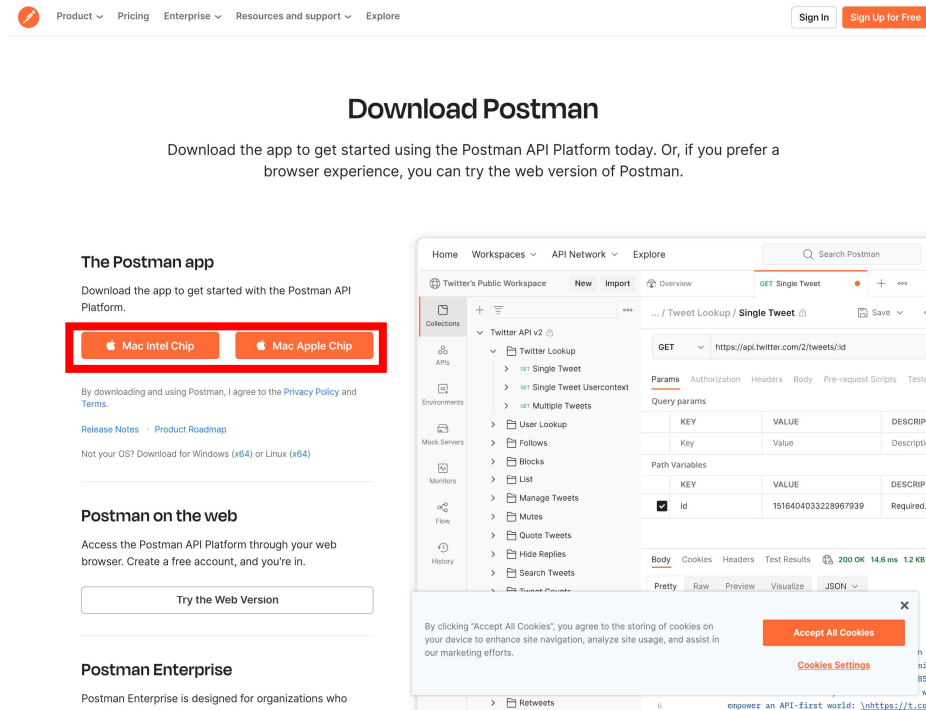
리소스뷰어 설치가 완료되면 위와 같이 실행할 수 있다.



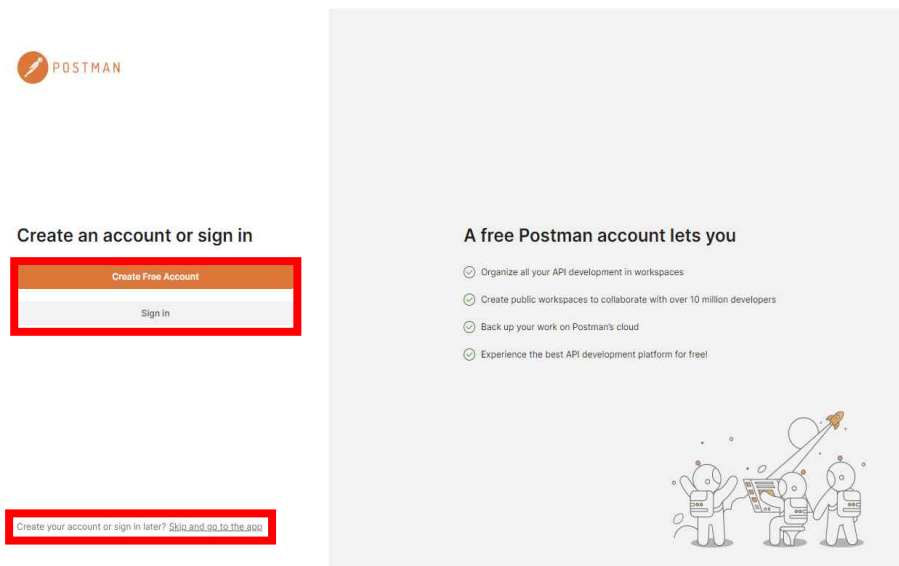
아래와 같이 리소스뷰어가 보이면 표신된 부분에 Mobius 에 생성된 리소스의 url 을 입력하고 Strat 버튼을 누르면 해당 url 의 리소스 구조를 볼 수 있다. url 부분의 처음은 콤보 박스로 되어 있고 이전 Mobius 와 Mobius release 2 를 선택할 수 있다. 이후 부분은 직접 입력해야 하고 여기에 생성한 AE 의 name 을 입력하면 된다.

3.2.2. Postman 활용

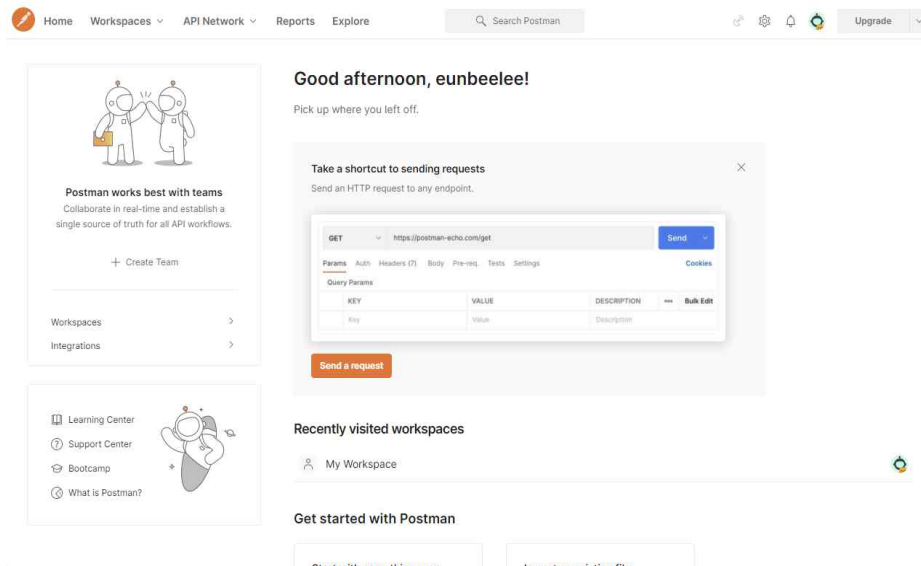
Postman 을 활용하여 직접 구축한 Mobius 를 동작을 테스트 할 수있다.



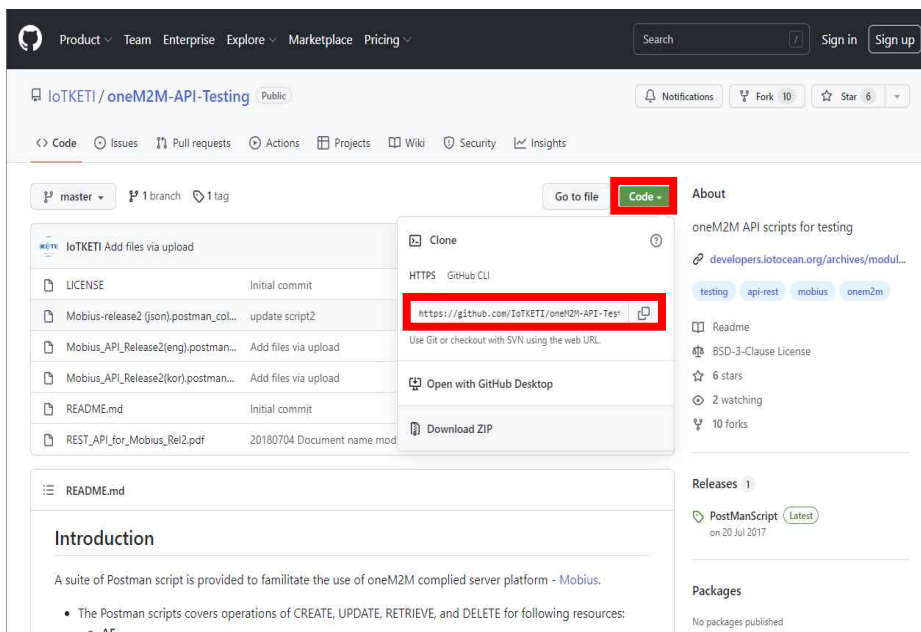
<https://www.postman.com/downloads/> 해당 사이트에 접속하여 postman 을 다운받을 수 있다.



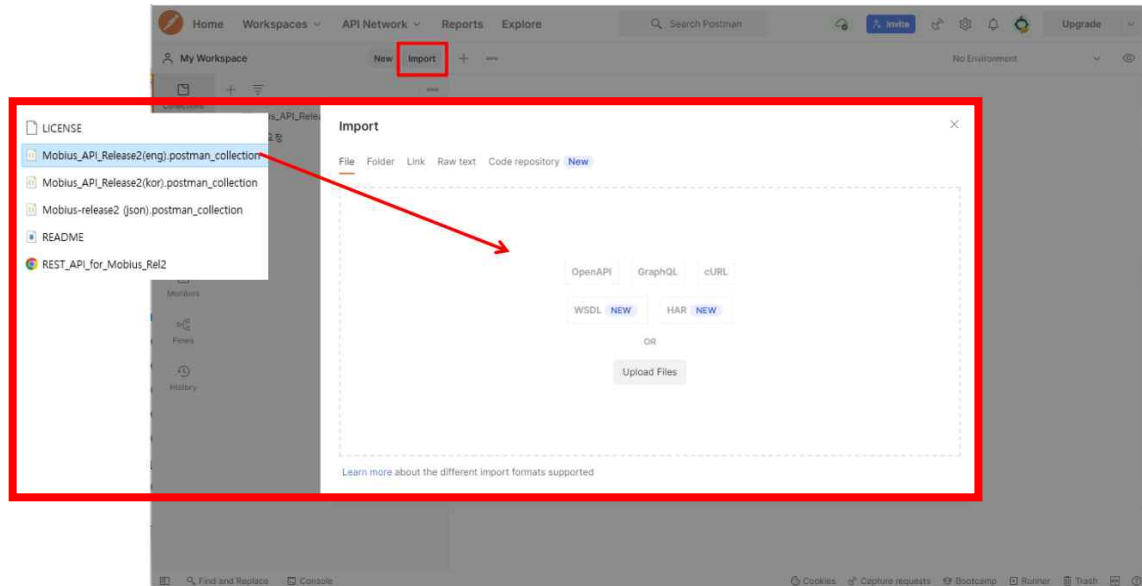
Postman 을 설치하면 회원가입 후 로그인한다.
로그인을 원하지 않으면 아래에 Skip 하면된다.



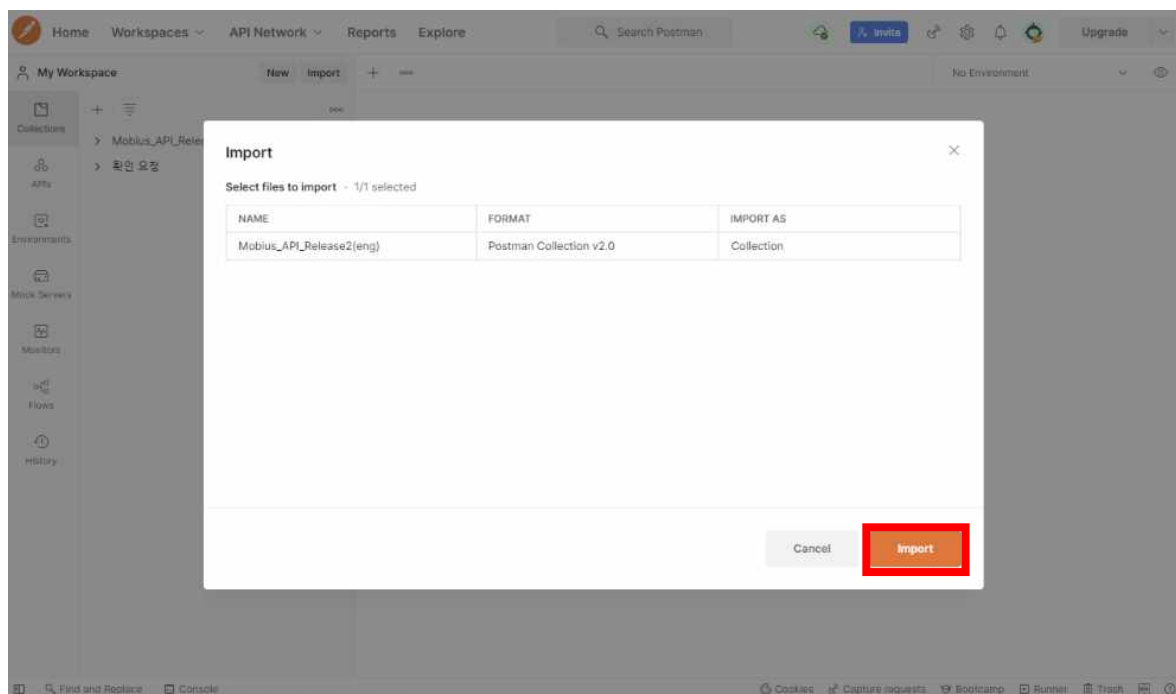
Postman 설치 후 로그인하면 위와 같이 실행된다.



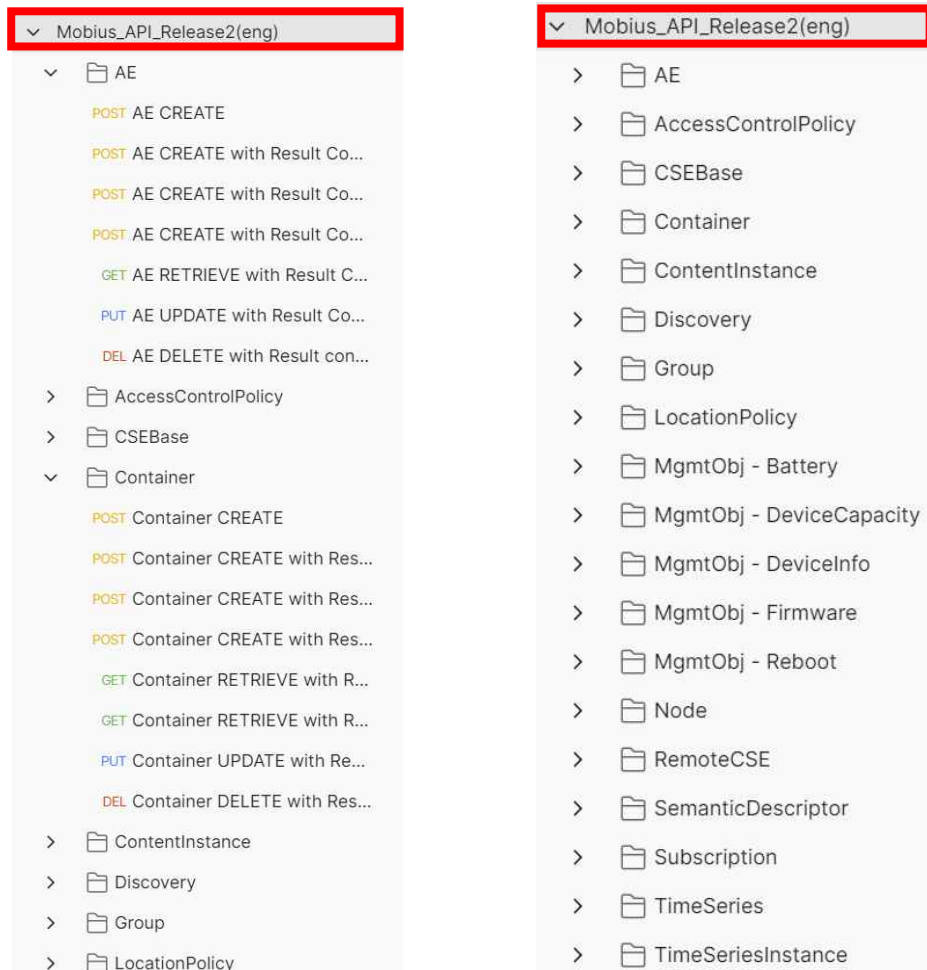
Mobius 테스트를 위해 OCEAN 에서 미리만들어 둔 스크립트를 다운받는다.



Import 하는 방법은 그림과 같이 My Workspace 옆에 Import 버튼을 선택한다.
Import 를 위한 창이 열리면 OCEAN 에서 다운로드 받은 스크립트 파일을 드래그한다.



Import 버튼을 클릭하여 import 를 완료한다.

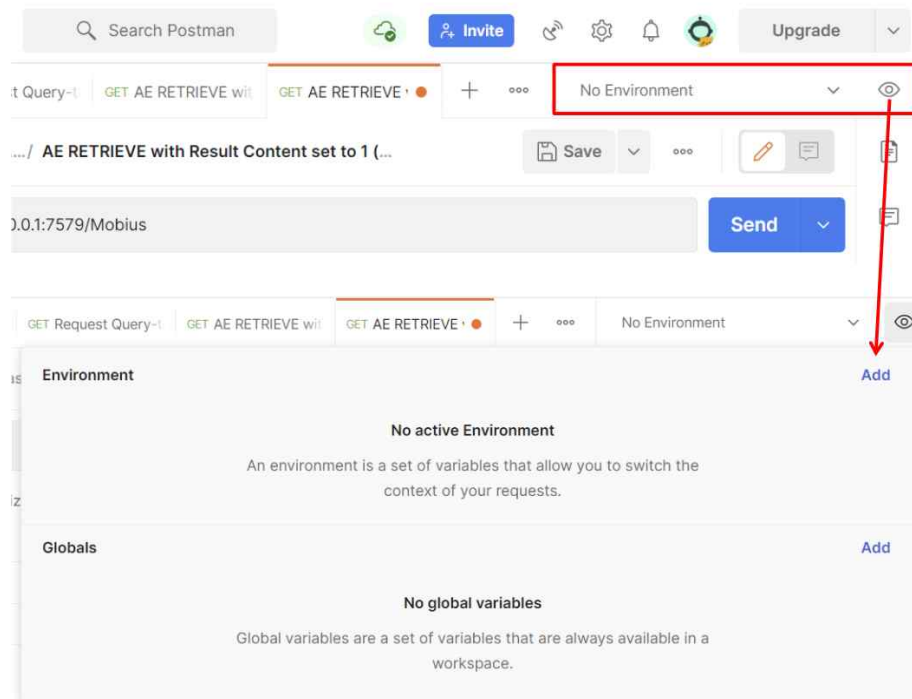


Import 가 성공하면 위의 그림처럼 Mobius_API_Release2(eng)이라는 컬렉션 탭이 생성이 될 것이다. 이 테스트 스크립트는 현재 지원되는 리소스의 CRUD 를 할 수 있도록 정리를 해 둔 것이고 xml 데이터 포맷으로 테스트 할 수 있도록 설정되어 있다.



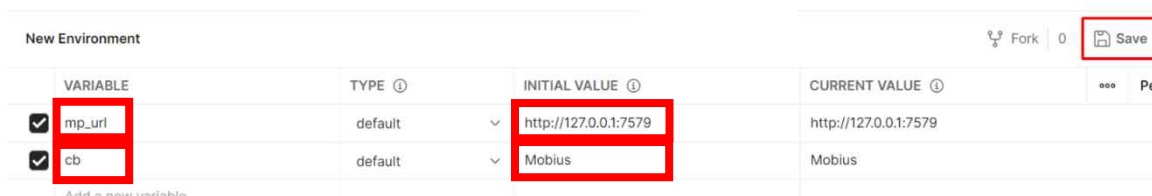
테스트를 하기 위해서는 Mobius 가 구동중인 서버의 IP 를 입력해야 한다.

위의 그림과 같이 주소창에서 값을 설정해 주어야 한다. {{mp_url}}, {{cb}} 대신에 http://(직접구축한서버주소):포트번호/csebase 값을 입력하여 사용할 수 있다.



{{mp_url}} 설정하는 방법은 위와 같다. 포스트맨 오른쪽 위에 보면 환경을 설정할 수 있는 리스트박스가 있고 이를 선택한다.

Environment 탭에서 Add 를 선택한다.



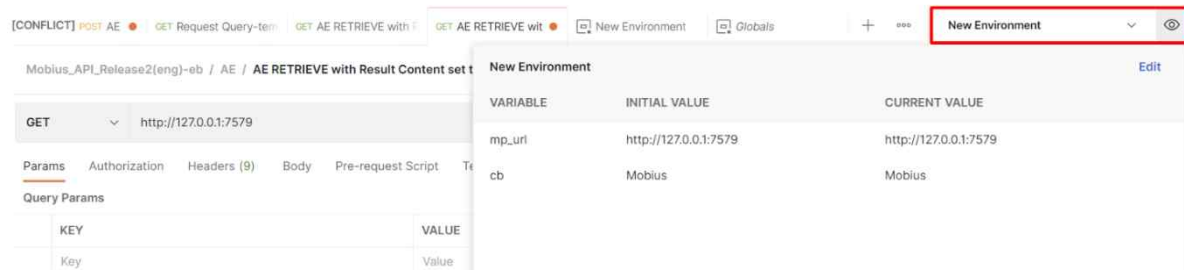
설정 창이 나오면 아래와 같이 {{mp_url}}, {{cb}} 값을 입력한다.

① VARIABLE 이름에 mp_url, cb 를 입력한다.

② mp_url 의 INITIAL VALUE 에 http://127.0.0.1:7579 를 입력한다. CURRENT VALUE 는 자동으로 입력된다.

③ cb 의 INITIAL VALUE 에 Mobius 를 입력한다. CURRENT VALUE 는 자동으로 입력된다.

설정이 완료되면 Save 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.