



## **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 OCEAN의 오픈 소스 발전에 공헌할 수 있다. 그러나, 개발된 소스코드가 OCEAN에 공헌되기를 원한다면 이 코드는 OCEAN의 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. Mobius란?.....	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.....	26
2.2.2. Node JS 설치.....	32
2.3. Mobius 설치.....	38
2.3.1. 다운로드 및 모듈 설치.....	38
2.3.2. mobiusdb 데이터베이스 생성 .....	41
3. Mobius 서버 플랫폼 구동.....	47
3.1. 환경설정.....	47
3.2. 테스트 .....	49
3.2.1. 리소스뷰어(oneM2M Browser) 활용 .....	49
3.2.2. Postman 활용.....	52

# 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.

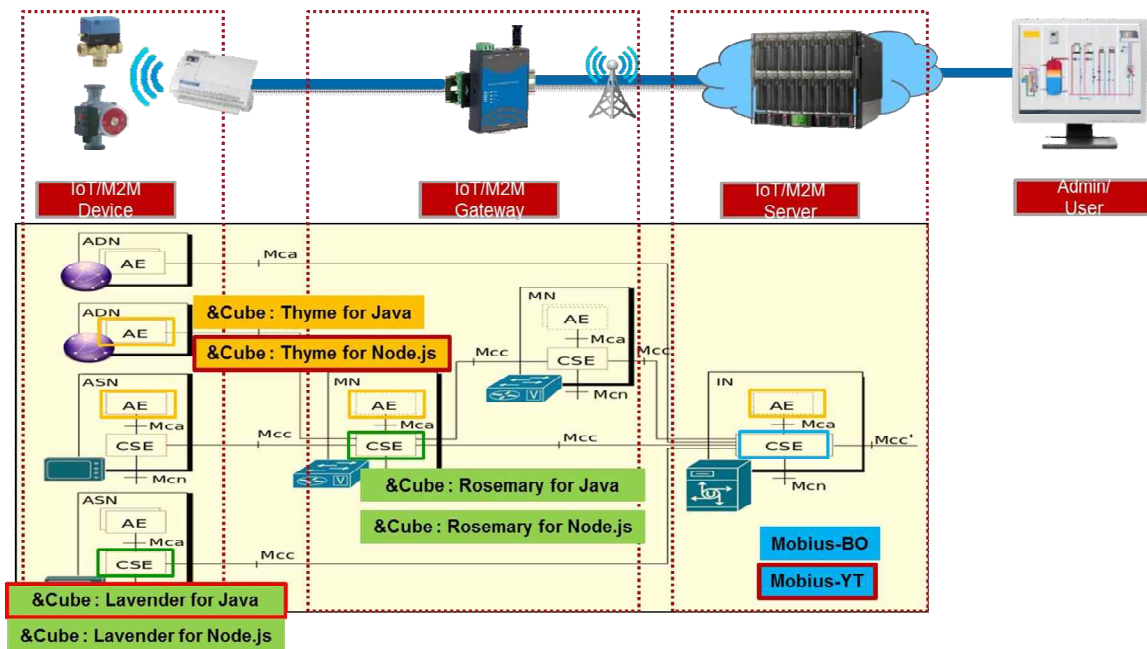


Figure 1 oneM2M 기반 Mobius 플랫폼 개념도

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

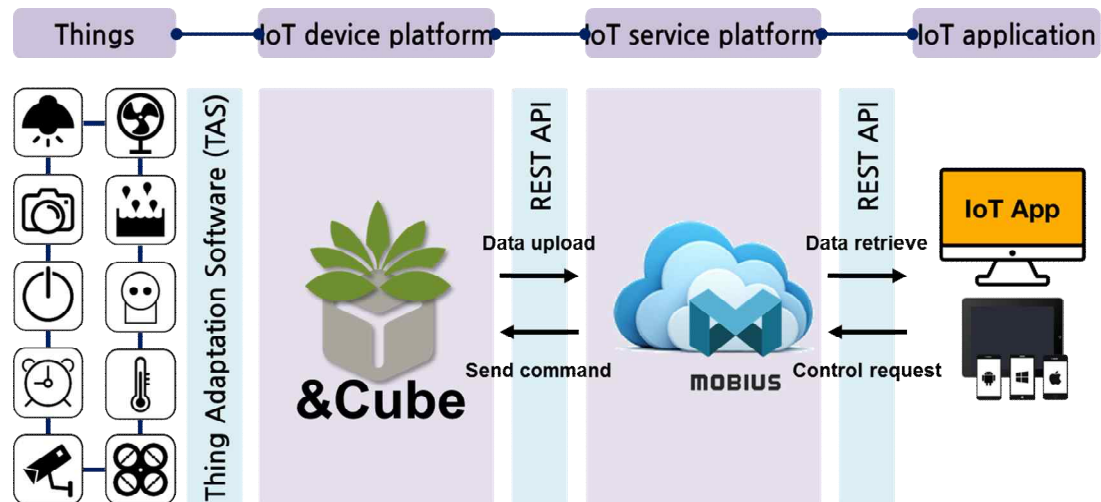


Figure 2 Mobius 와 IoT 디바이스 연동

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 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.

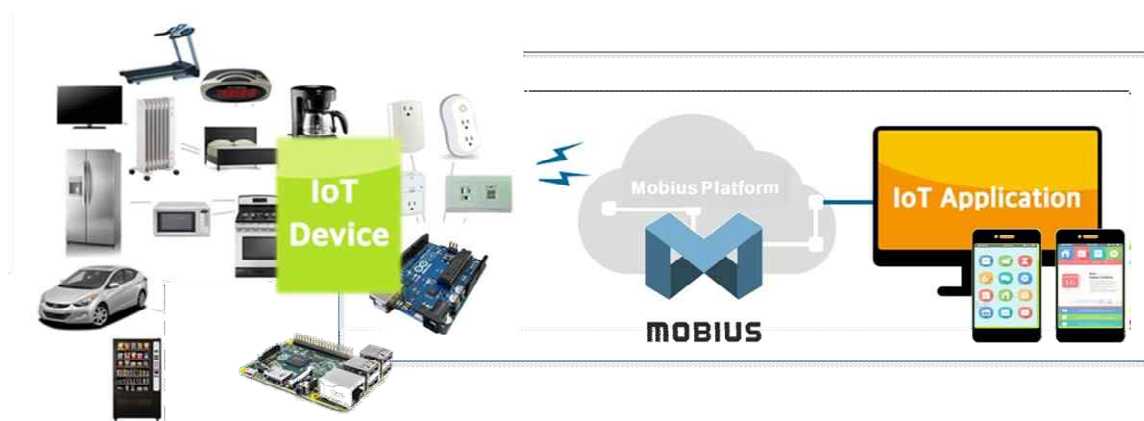


Figure 3 Mobius Platform 개요

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

Mobius Platform은 oneM2M 표준을 준수하여 Http, MQTT, COAP, WebSocket 프로토콜과 Open API를 제공한다. oneM2M 표준에 따라 Mobius의 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



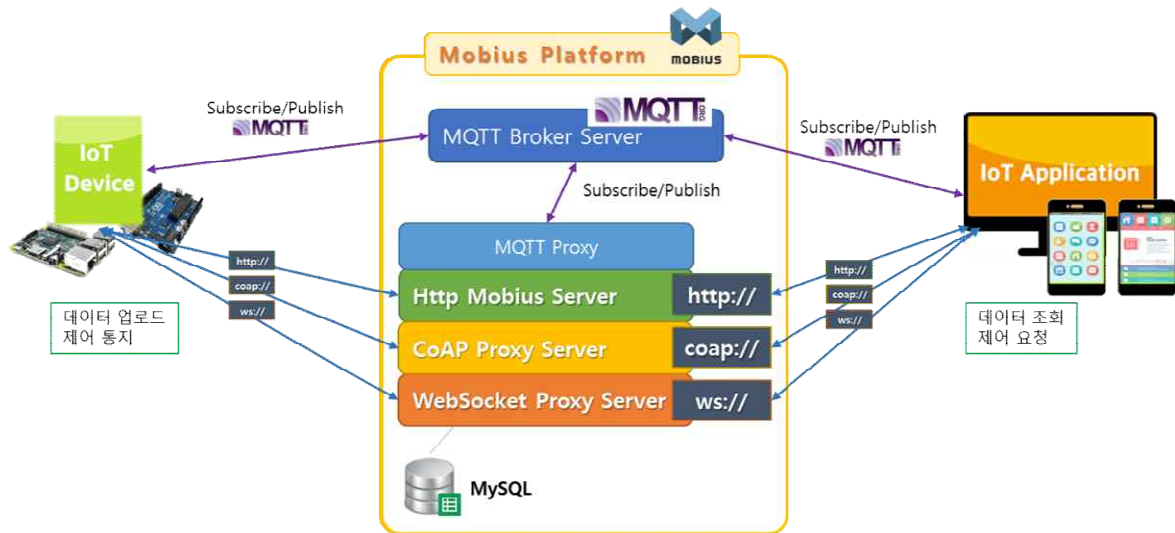


Figure 5 Mobius 서버 플랫폼 시스템 구성

### 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.

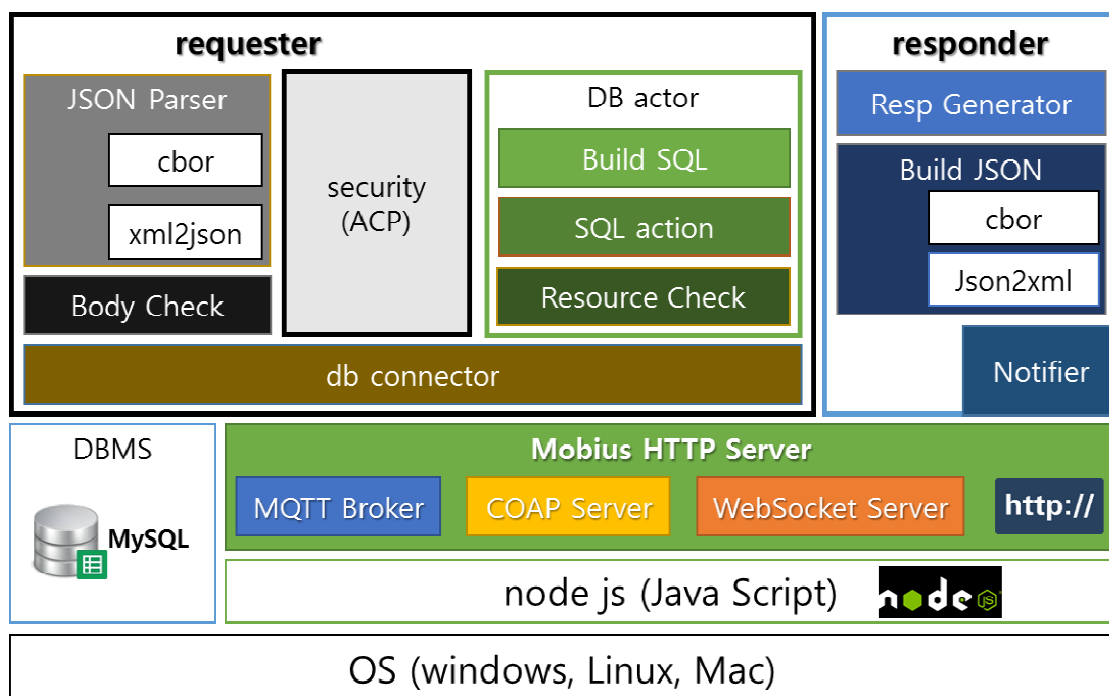


Figure 6 Mobius S/W 아키텍처

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

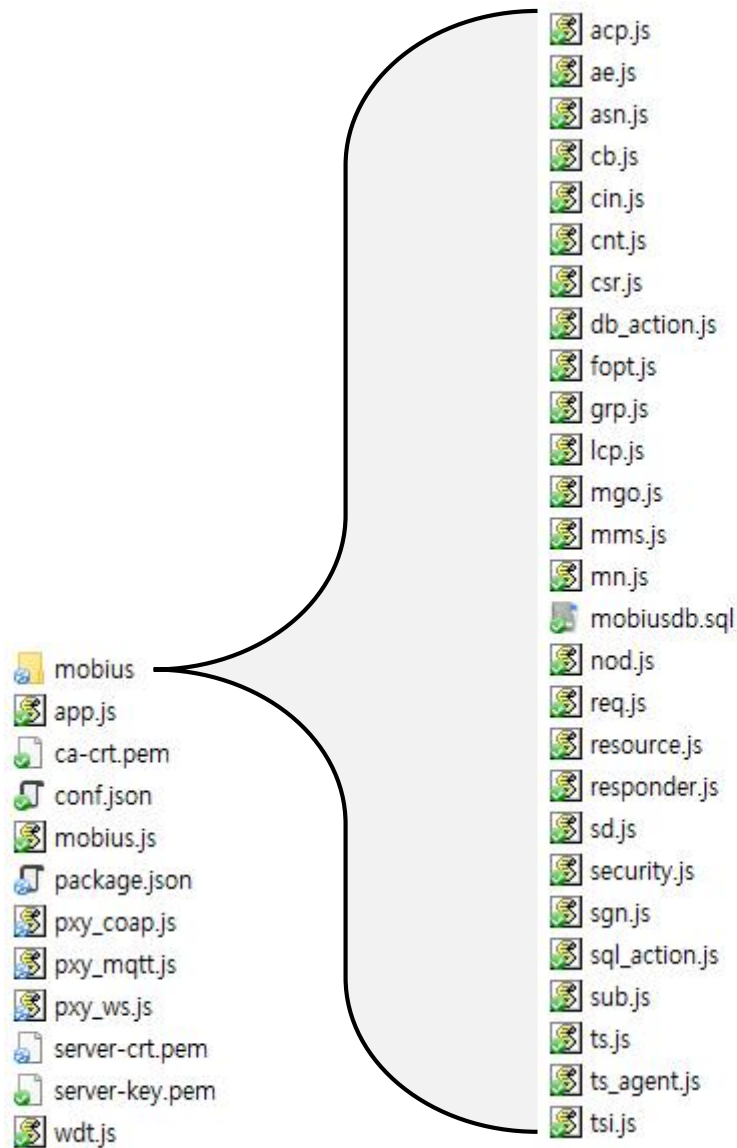
[http://hostname:port/IN-CSE\\_name/AE\\_name/container\\_name/contentInstance\\_name](http://hostname:port/IN-CSE_name/AE_name/container_name/contentInstance_name)

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 이 중에서 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 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 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 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 얻기 위한 sql문을 만들고 이를 db_action.js의 함수를 통해 데이터베이스로부터 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 체크하여 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 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 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 (Windows)

#### 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.

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

MySQL에 들어가 보면 MySQL의 다양한 버전이 있는데 위 링크를 클릭해서 들어가면 무료로 사용할 수 있는 GPL버전인 MySQL Community Server를 다운받을 수 있다.

General Availability (GA) Releases Archives

## MySQL Community Server 8.0.31

Select Operating System:  
Microsoft Windows

[Looking for previous GA versions?](#)

**Recommended Download:**


### MySQL Installer for Windows

All MySQL Products. For All Windows Platforms. In One Package.

Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.


**Windows (x86, 32 & 64-bit), MySQL Installer MSI**

[Go to Download Page >](#)



**Other Downloads:**

Download	Version	Size	Action
Windows (x86, 64-bit), ZIP Archive (mysql-8.0.31-winx64.zip)	8.0.31	222.3M	<a href="#">Download</a>
Windows (x86, 64-bit), ZIP Archive Debug Binaries & Test Suite (mysql-8.0.31-winx64-debug-test.zip)	8.0.31	555.6M	<a href="#">Download</a>

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

## MySQL Installer 8.0.31

Select Operating System:  
Microsoft Windows

[Looking for previous GA versions?](#)

Download	Version	Size	Action
Windows (x86, 32-bit), MSI Installer (mysql-installer-web-community-8.0.31.0.msi)	8.0.31	5.5M	<a href="#">Download</a>
Windows (x86, 32-bit), MSI Installer (mysql-installer-community-8.0.31.0.msi)	8.0.31	431.7M	<a href="#">Download</a>

여기서는 윈도우에 MySQL을 설치하기 위해 Windows (x86, 64-bit), MySQL Installer MSI라고 적힌 Download 버튼을 클릭한다. (MySQL에는 MySQL Enterprise Edition이나 MySQL Cluster CGE처럼 유료로 사용해야 하는(commercial) 것도 있으니 참고하길 바란다.)

온라인 설치가 아닌 오프라인으로 설치하기 위해 아래의 용량이 큰 설치파일을 다운로드 한다. 이 버전은 웹 설치가 어려운 곳에서 MySQL을 설치할 때 이 버전을 활용할 수 있다.



## 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 »

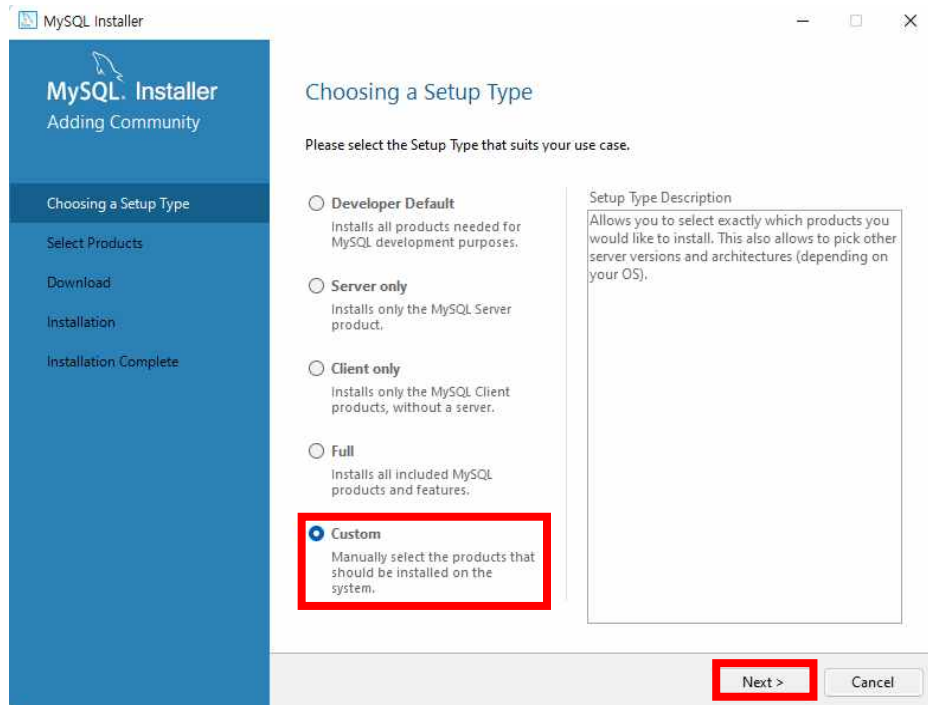
for an Oracle Web account

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

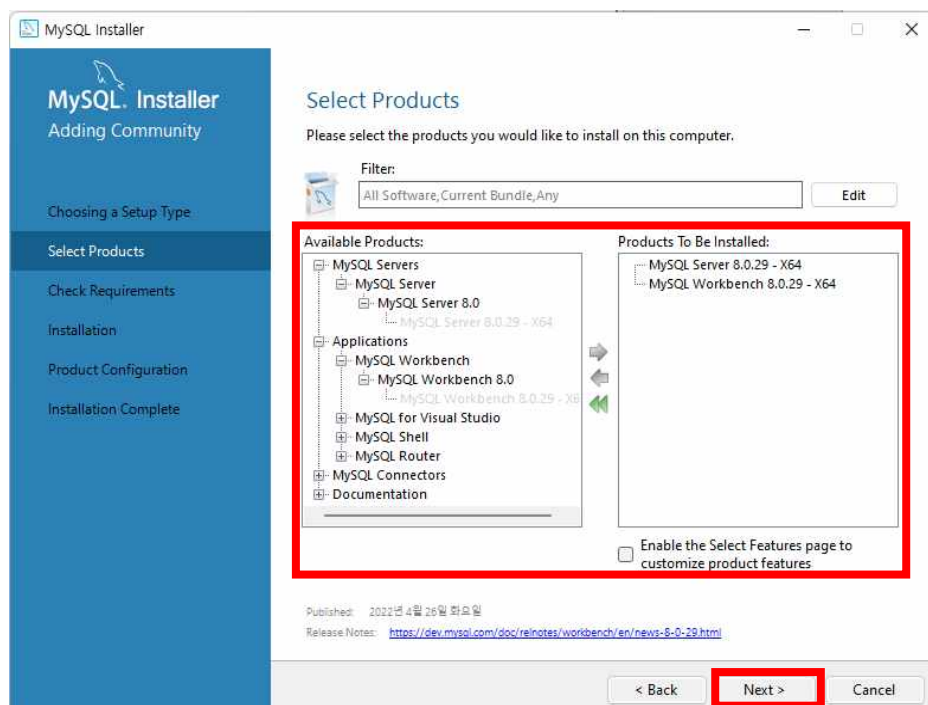
**No thanks, just start my download.**

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

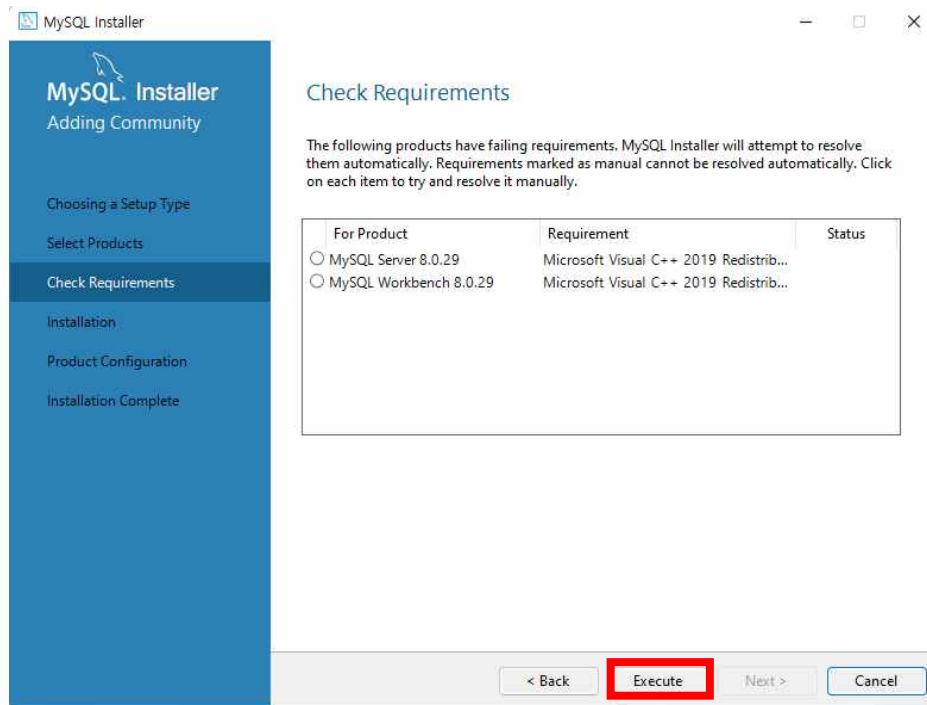
MySQL 설치파일을 다운받아서 실행시켜 라이선스에 동의를 하고 Next 버튼을 누른다.



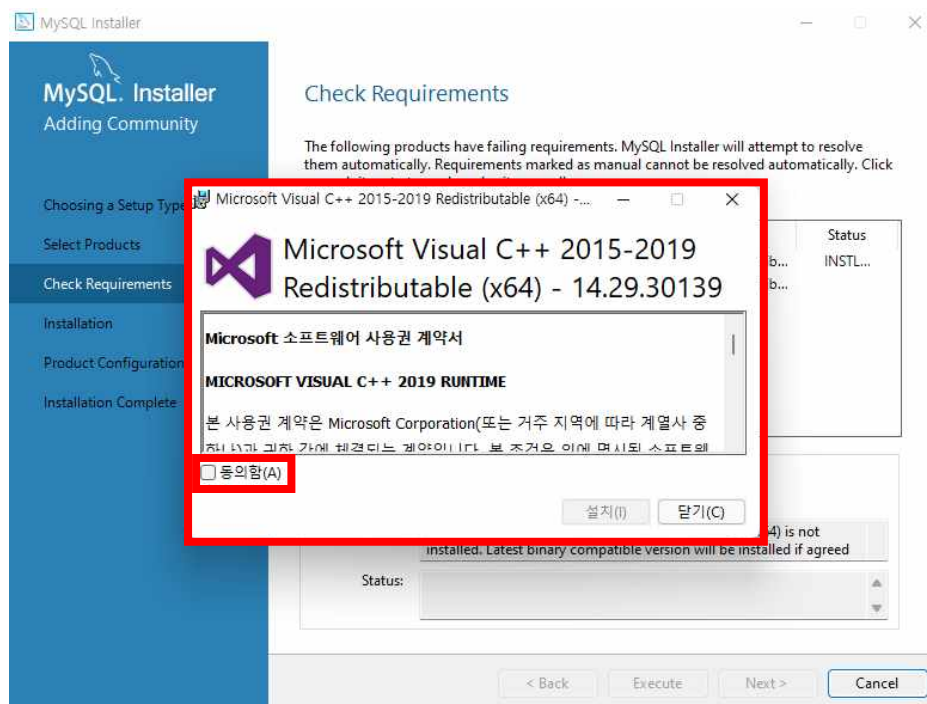
설치 스타일에는 Developer Default, Server only, Client only, Full, Custom 이렇게 5가지가 있는데, Mobius를 위해서는 MySQL Server와 MySQL Workbench가 필요하다. 여기서 Custom을 선택하고 Next 버튼을 누르면서 설치를 진행한다.



MySQL Server와 MySQL Workbench를 선택한다. 선택 후 Next를 클릭한다.

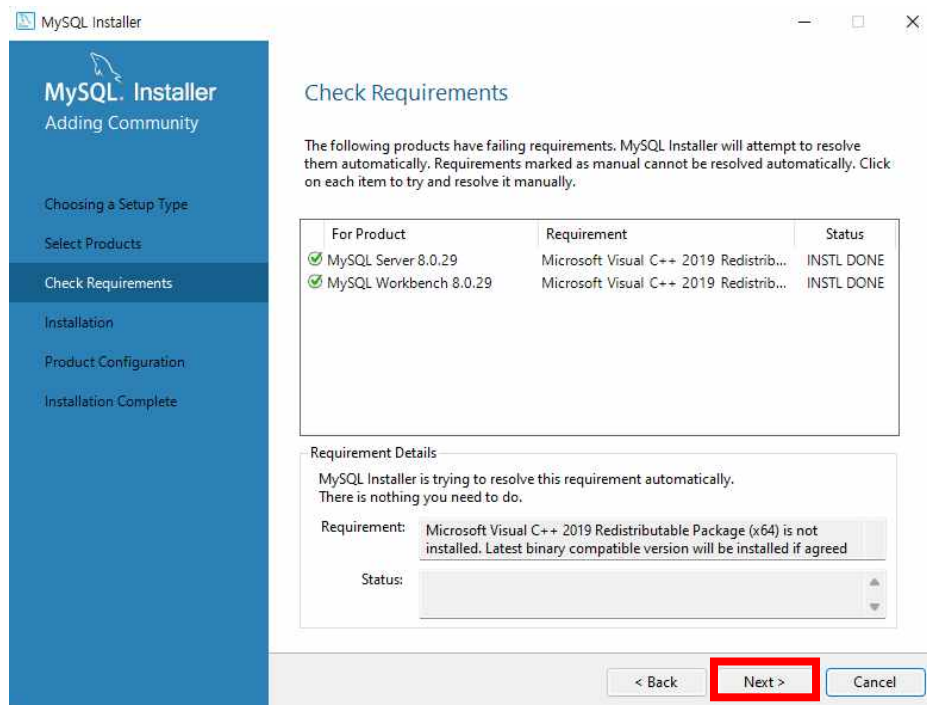


MySQL Server와 MySQL Workbench를 설치하기 위해 Execute를 클릭한다.

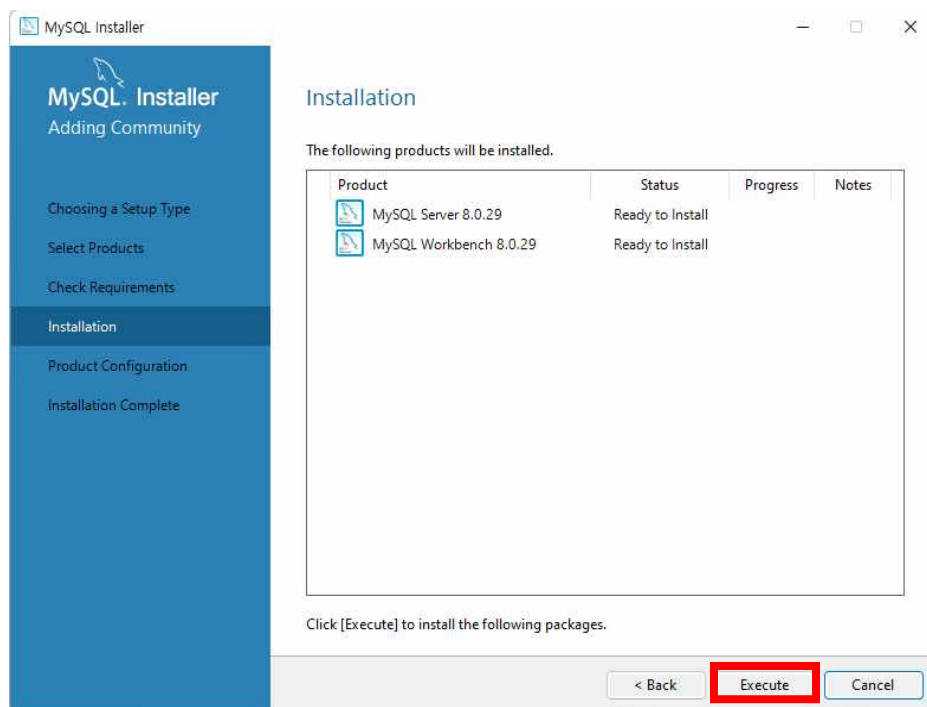


Visual C++가 설치되어 있지 않으면 위와 같이 C++ 런타임 구성요소를 설치하기 위한 창이 나타나는 데 라이선스 동의 후 설치를 진행하면 된다.

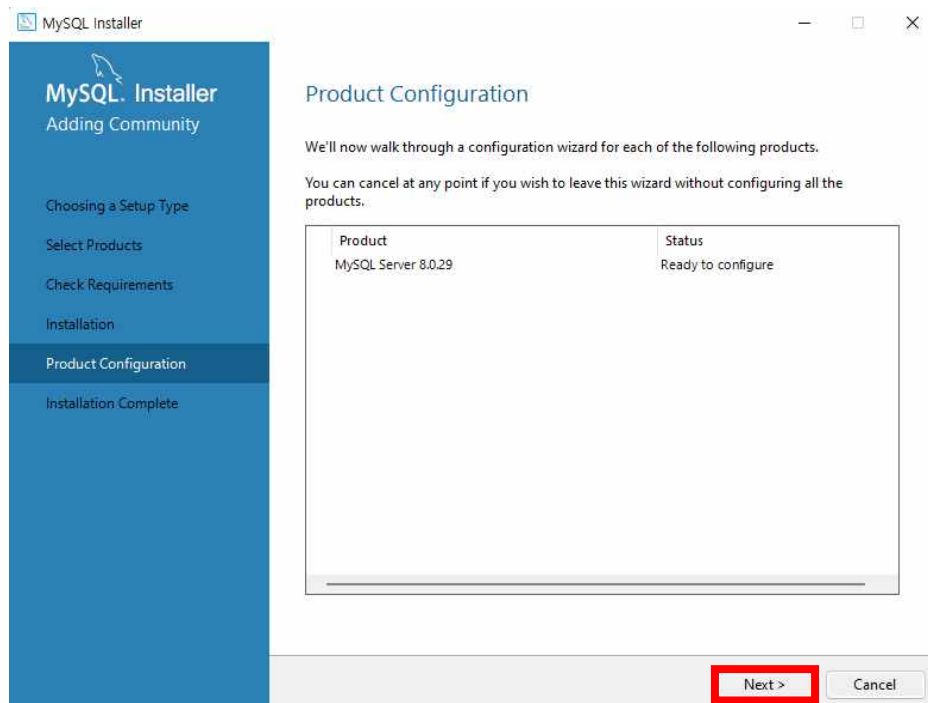
이미 Visual C++가 설치되어 있으면 위 창이 나타나지 않으므로 넘어가면 된다.



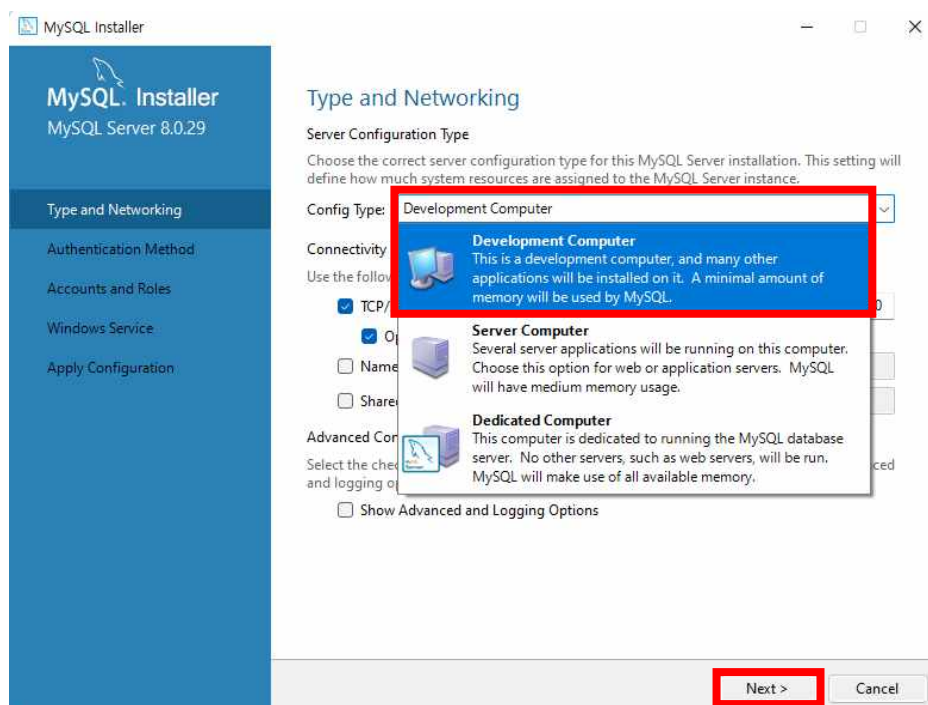
설치를 위해 Next를 클릭한다.



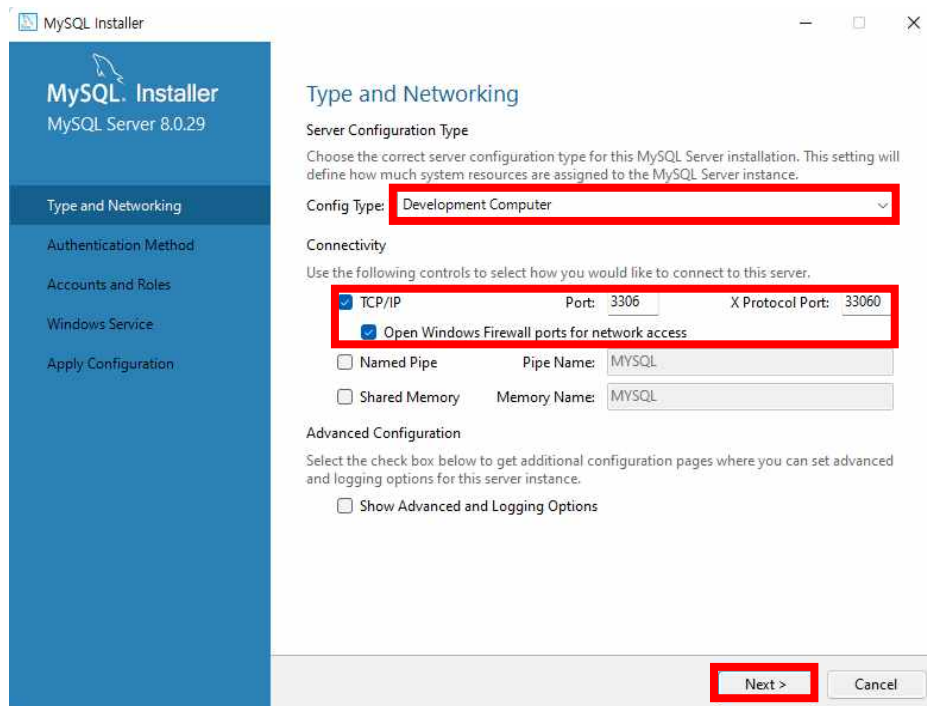
설치를 위해 Execute를 클릭한다.



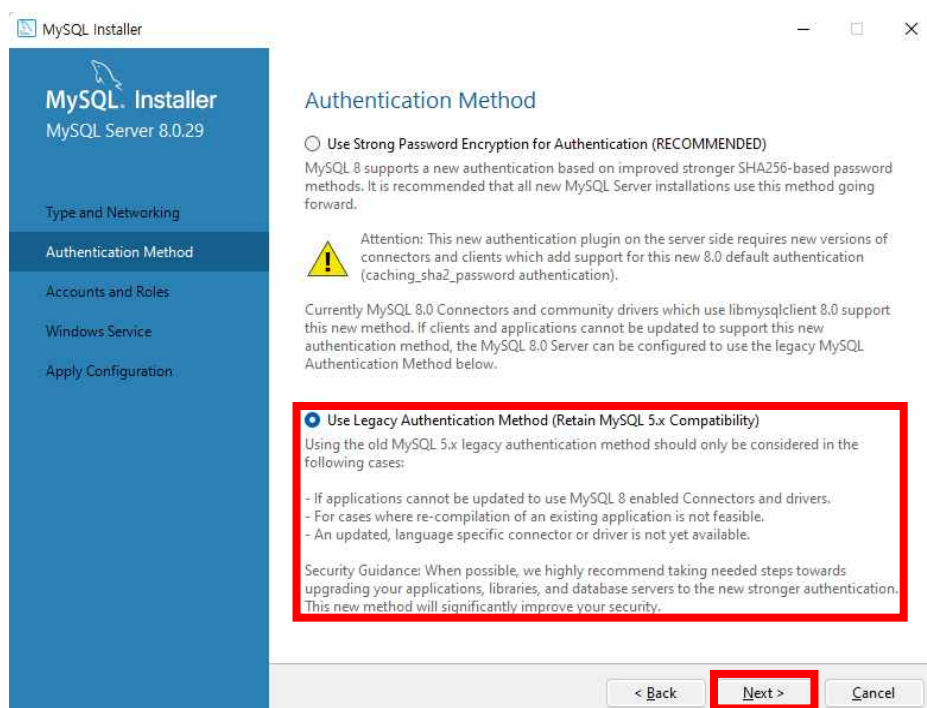
MySQL 설치가 완료되면 환경설정 작업을 위해 Next를 클릭한다.



환경설정 타입에는 Development Machine, Server Machine, Dedicated Machine이 있는데 필요한 것을 선택하면 되고 여기서는 개발자라고 가정하고 Development Machine을 선택한다. Next를 클릭한다.

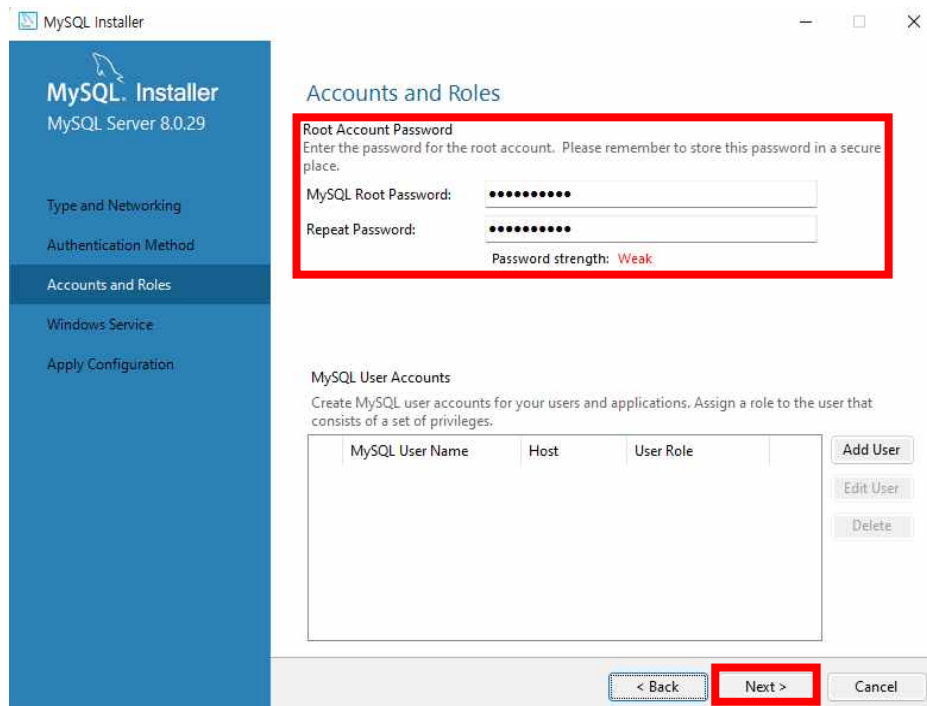


MySQL의 기본 포트는 3306이다. Next를 클릭한다.



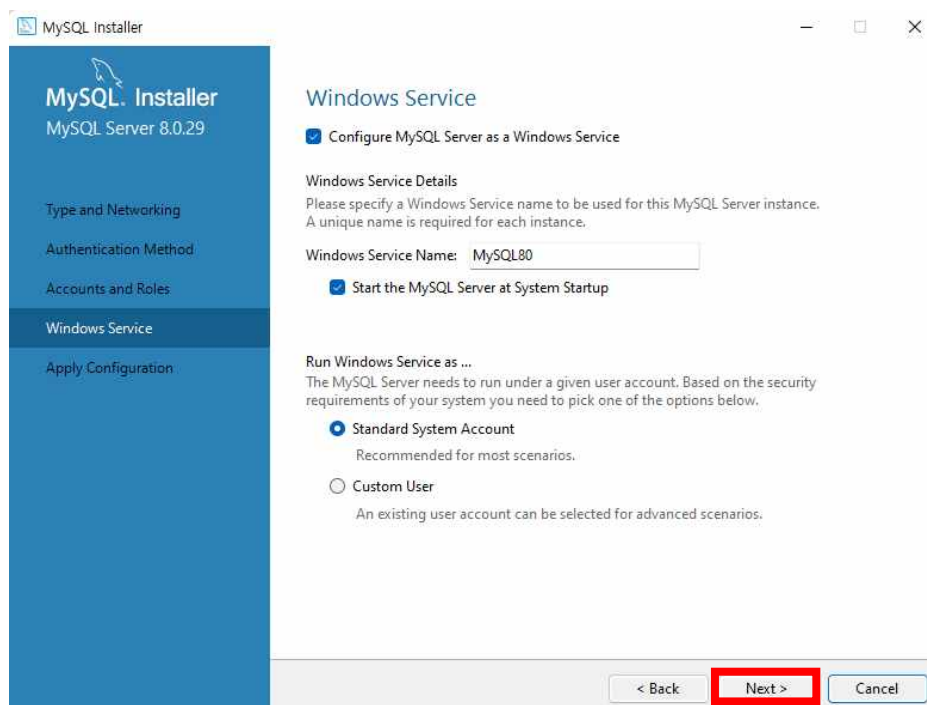
MySQL 8.0부터 새로운 사용자 인증 방식이 적용되었는데 Mobius 연동 및 기존 프로그램들과 호환을 위해 꼭 **Legacy Authentication Method**로 설치를 진행하여야 한다.



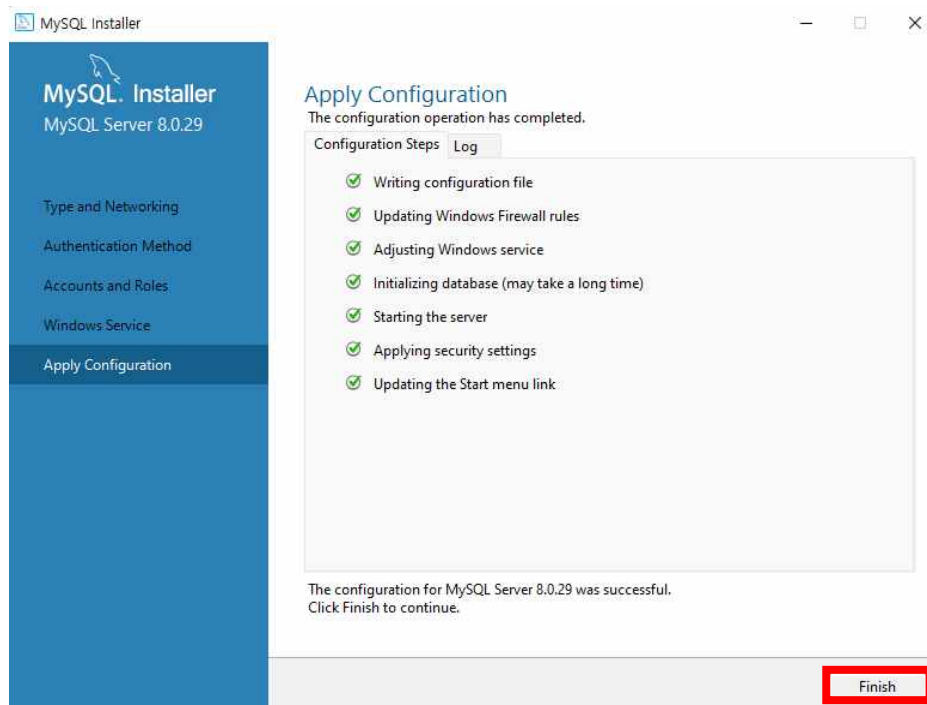


루트암호(Root Password)를 지정한 후 Next를 클릭한다.

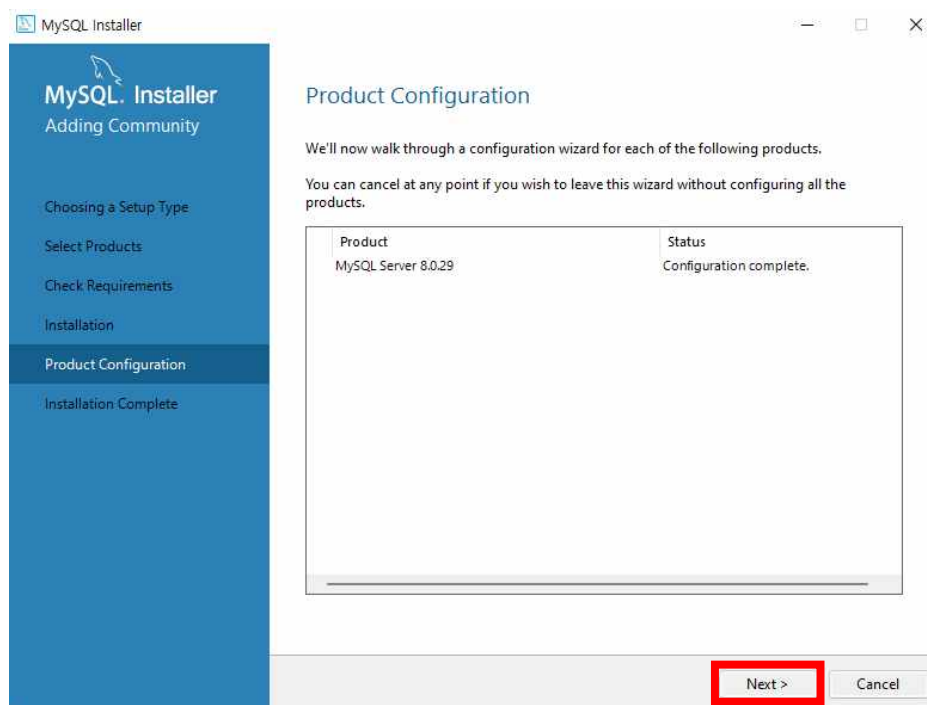
Mobius와 Mysql을 연동할 때 필요하므로 꼭 기억하여야한다.



윈도우 부팅 시 MySQL 자동 실행 옵션을 설정 할 수 있는 화면으로 윈도우 서비스 이름을 지정한다. Next를 클릭한다.

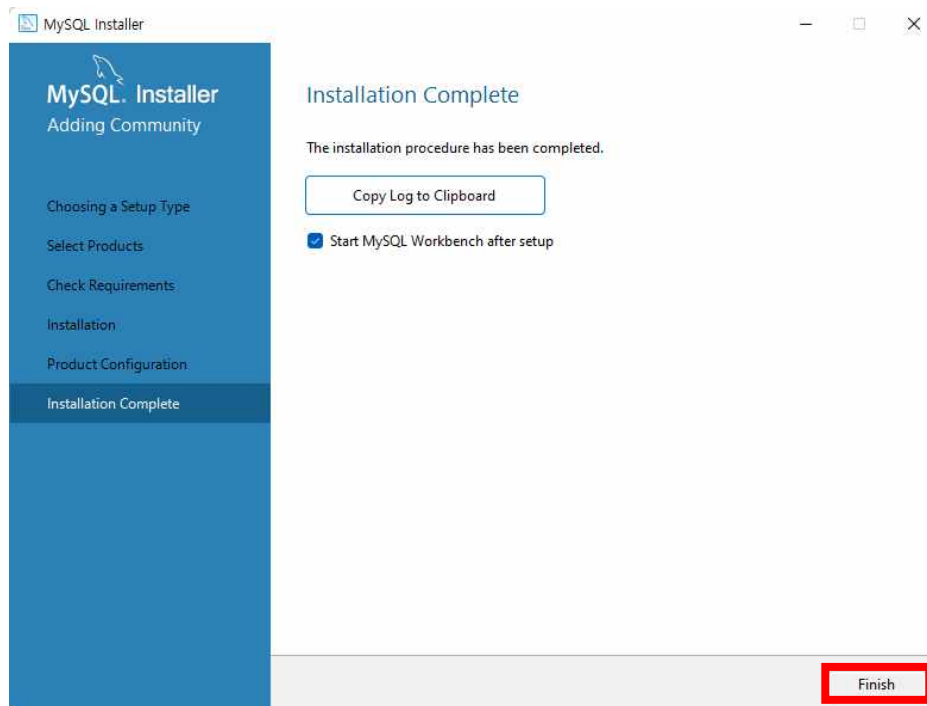


MySQL Server 환경설정이 제대로 설정된 것을 확인하고 Finish를 누른다.



기타 환경설정을 위해 Next를 클릭한다.





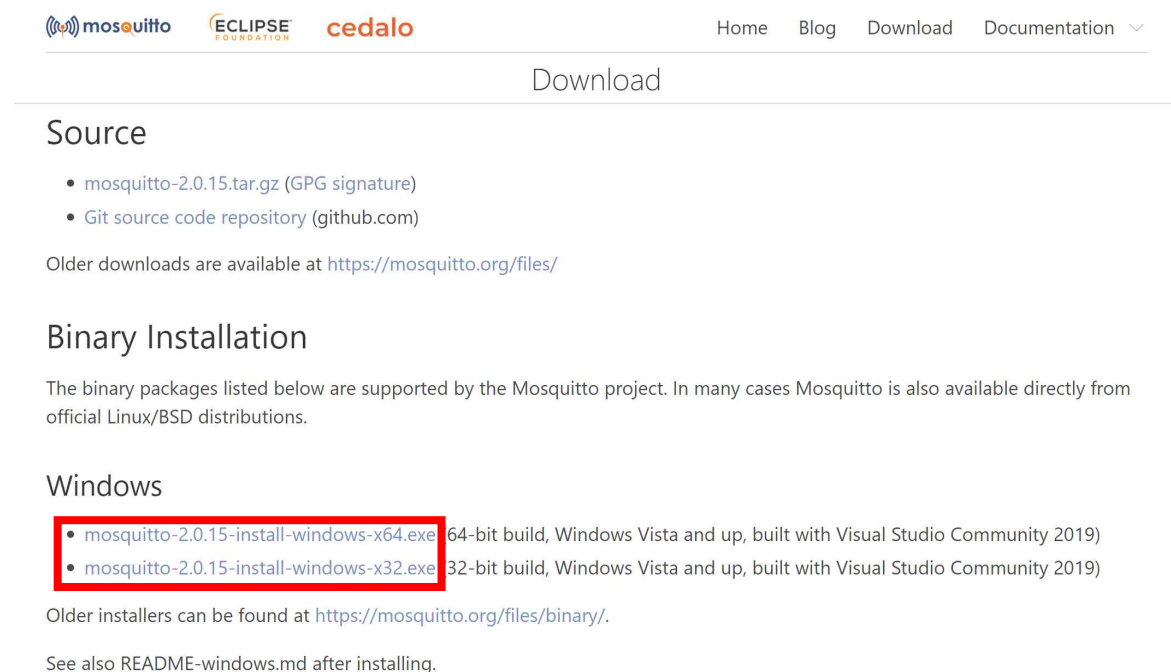
MySQL 설치가 완료되었다. 마지막 Finish 버튼을 누른다.

## 2.2.1. MQTT 브로커 설치

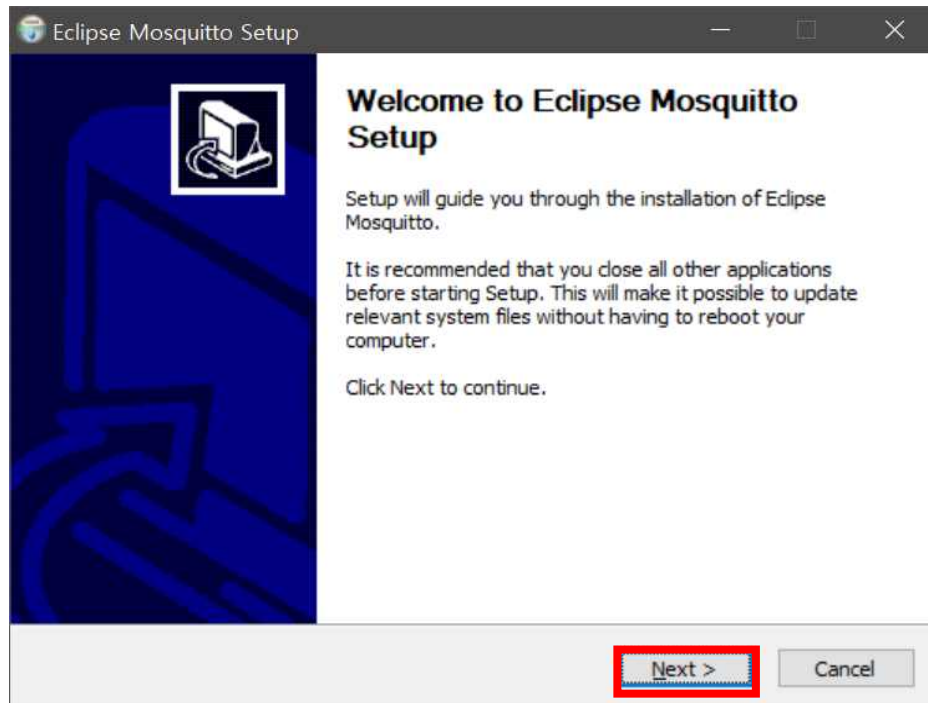
MQTT(Message Queuing Telemetry Transport)는 발행-구독 기반의 메시징 프로토콜이다. Mobius Platform은 디바이스와의 통신을 위해 MQTT 프로토콜을 지원하고 있으며 이를 위해서 오픈소스로 제공되는 MQTT 브로커인 Mosquitto 서버를 설치한다.

<https://mosquitto.org/download/>

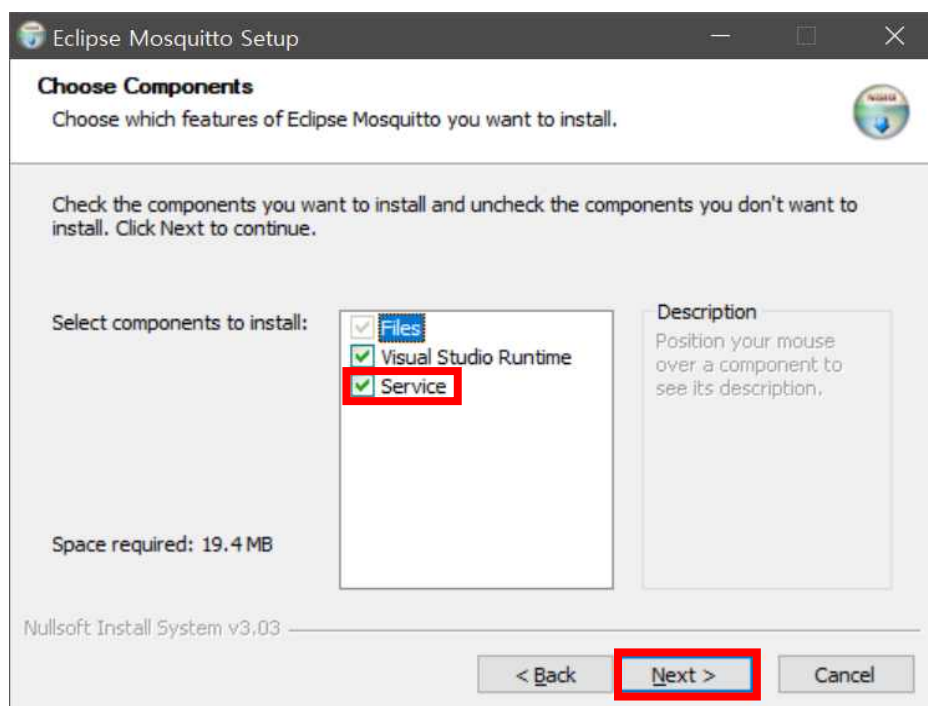
사이트에서 다운로드 메뉴를 누르고 Windows 설치파일 ( [mosquitto-2.0.15-install-windows-x64.exe](#) )을 선택하여 mosquitto 설치파일을 다운로드 하여 설치한다.

The screenshot shows the Mosquitto download page. At the top, there are logos for mosquitto, Eclipse Foundation, and cedalo, along with navigation links: Home, Blog, Download, and Documentation. The main heading is "Download". Under the "Source" section, there are two links: "mosquitto-2.0.15.tar.gz (GPG signature)" and "Git source code repository (github.com)". Below this, it says "Older downloads are available at https://mosquitto.org/files/". The "Binary Installation" section states that the listed packages are supported by the Mosquitto project. Under the "Windows" section, there are two links: "mosquitto-2.0.15-install-windows-x64.exe" (64-bit build, Windows Vista and up, built with Visual Studio Community 2019) and "mosquitto-2.0.15-install-windows-x32.exe" (32-bit build, Windows Vista and up, built with Visual Studio Community 2019). Below this, it says "Older installers can be found at https://mosquitto.org/files/binary/." and "See also README-windows.md after installing."

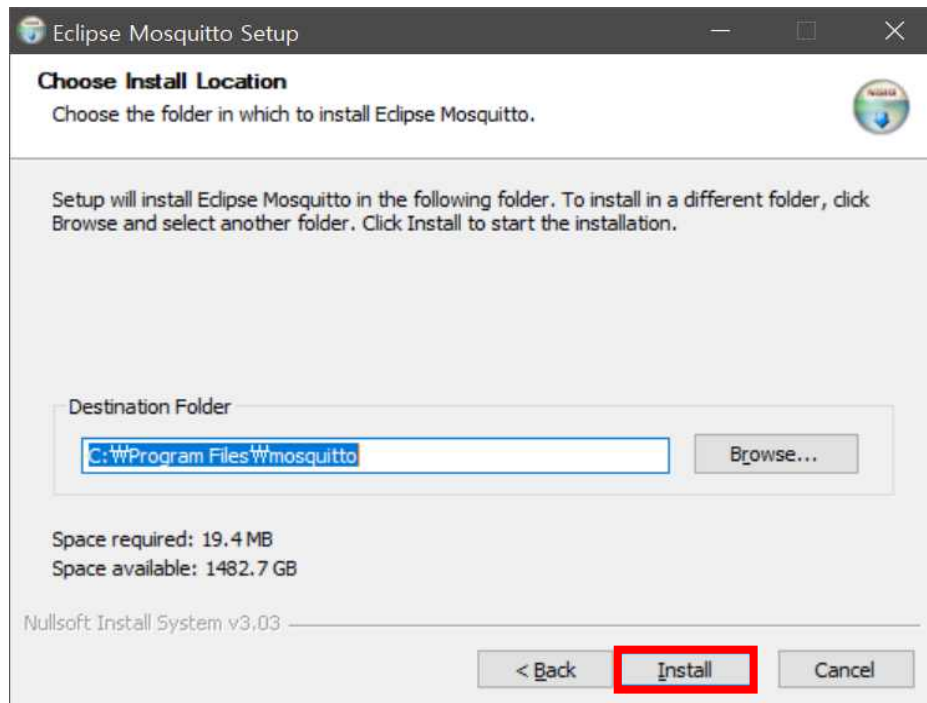
설치 환경의 운영체제에 맞게 설치파일을 다운로드 한다.



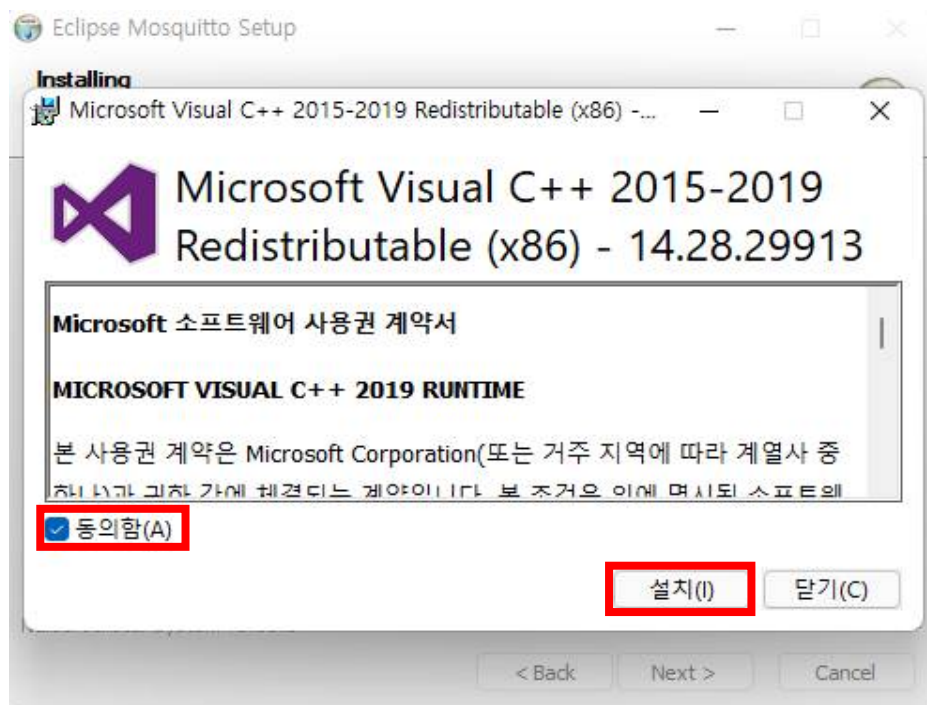
설치를 위해 Next를 클릭한다.



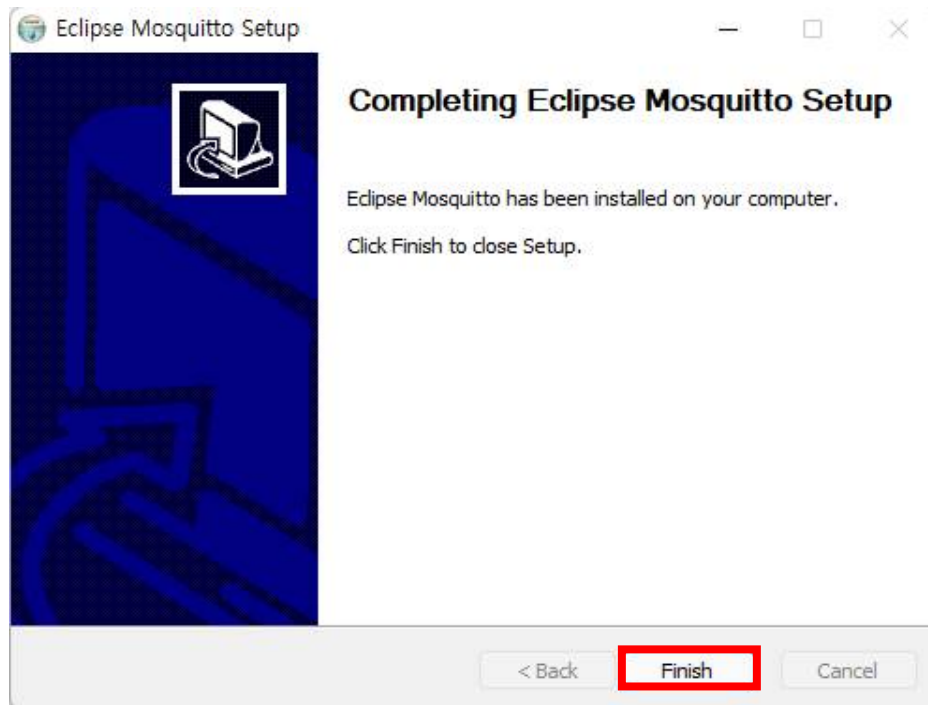
설치 중에 위의 그림처럼 mosquitto 서버를 윈도우 서비스로 시작하는 옵션이 나오는 데 위와 같이 디폴트로 두고 Next를 누른다. 설치 후 별도의 작업없이 mosquitto 서버가 윈도우 서비스로 등록되어 자동으로 시작되는 옵션이다.



설치를 위해 Install을 클릭한다.



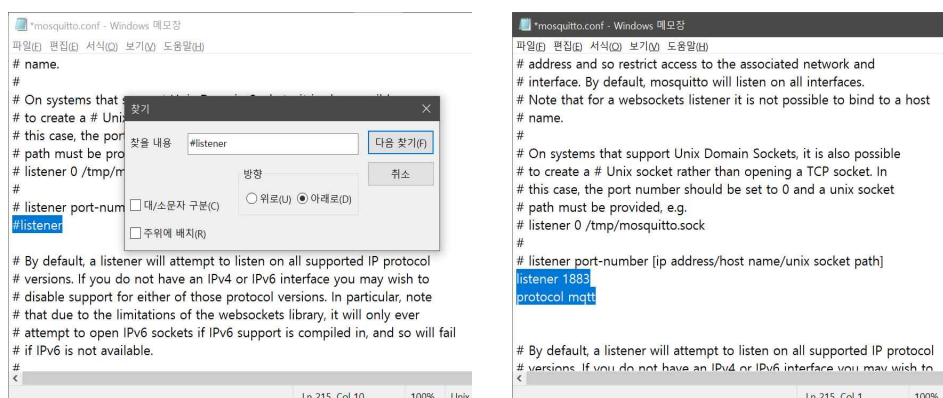
Visual C++가 설치되어 있지 않으면 위와 같이 C++ 런타임 구성요소를 설치하기 위한 창이 나타나는 데 라이선스 동의 후 설치를 진행하면 된다.



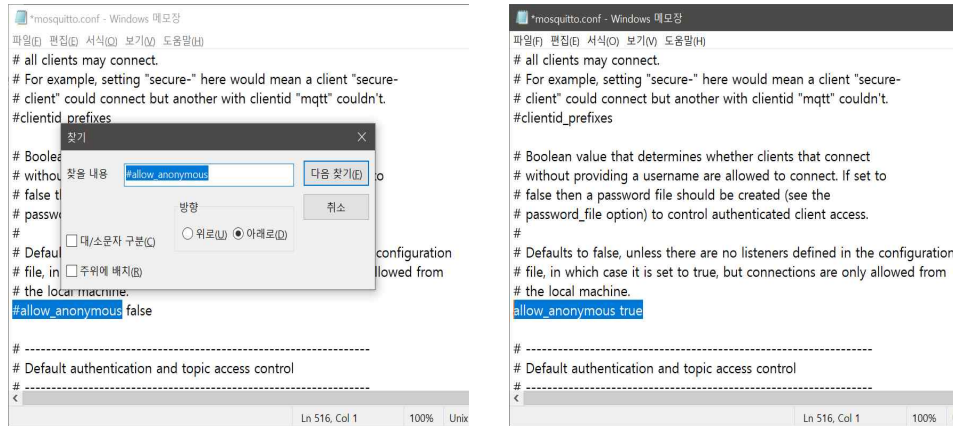
설치가 완료되면 Finish 버튼을 클릭하여 설치를 완료한다.

mosquitto는 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 mosquitto를 설치할 때 설정한 경로( C:\Program Files\Mosquitto )의 폴더 안에 있다. 설정 파일 수정을 위해 바탕화면으로 복사하고 복사한 설정 파일을 열어 아래와 같이 수정한다.



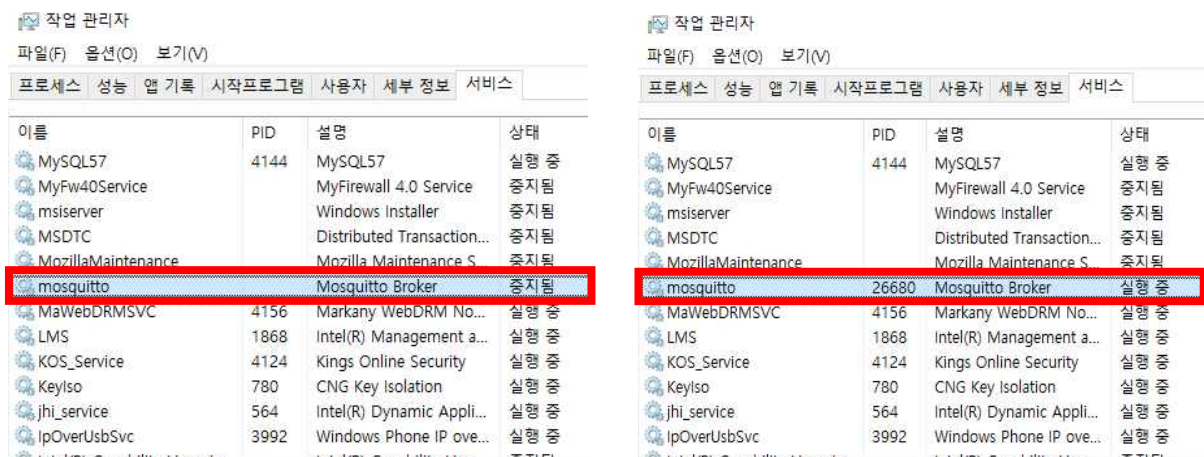
"#listener"를 검색하여 오른쪽 그림과 같이 "listener 1883", "protocol mqtt"으로 수정한다.



"#allow\_anonymous"를 검색하여 오른쪽 그림과 같이 "allow\_anonymous true"로 수정한다.

변경한 설정 파일을 저장 후 원래 경로로 옮겨준다. 여기서 관리자 권한을 필요로 하는데 계속 버튼을 클릭한다.

설치가 완료되면 윈도우 서비스로 등록한 mosquitto를 시작해 주어야 한다. 다음 그림과 같이 윈도우 작업관리자를 열면 서비스 탭에서 mosquitto를 찾을 수 있다.

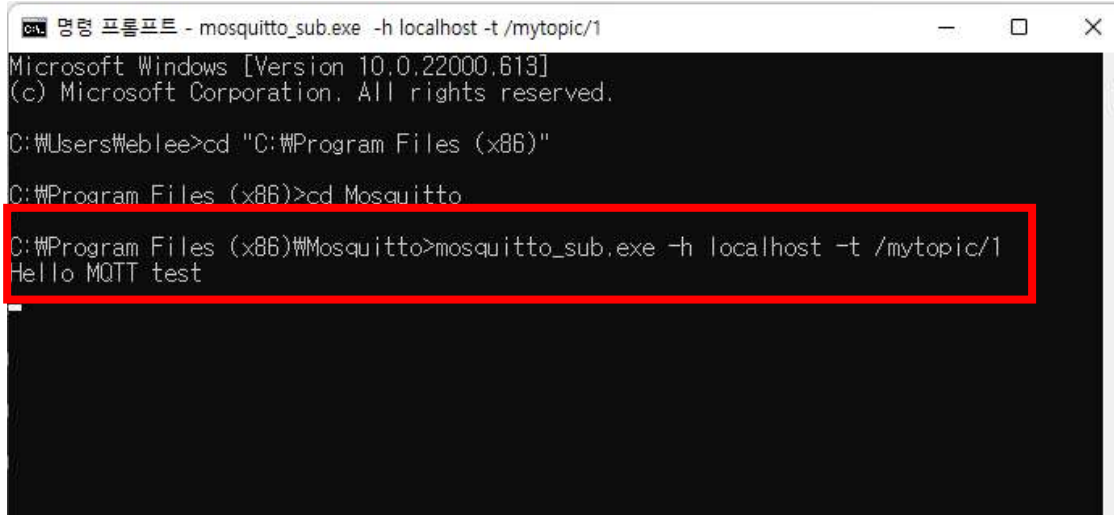


처음 설치된 후에 mosquitto는 중지됨으로 되어있을 것이고 마우스 오른쪽 버튼을 눌러 시작을 눌러 서비스를 시작할 수 있다.

mosquitto가 제대로 설치됐는지 테스트하기 위해 명령 프롬프트를 실행시키고 아래와 같이 mosquitto 서버가 설치된 폴더로 이동하여 다음의 명령어를 실행한다.

구독 (Subscribe)

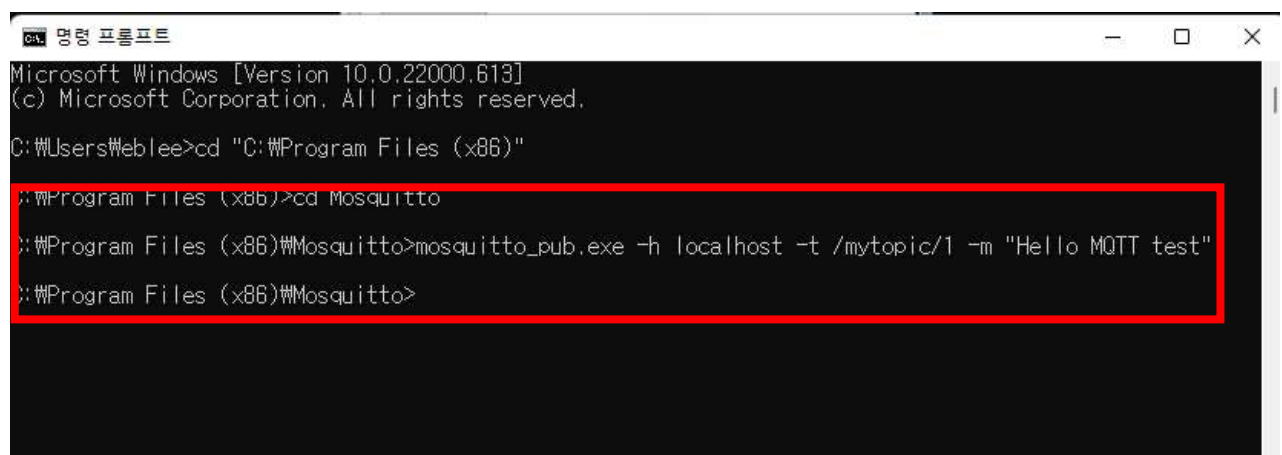
```
> mosquitto_sub.exe -h localhost -t /mytopic/1
```



```
C:\Users\Weblee>cd "C:\Program Files (x86)"
C:\Program Files (x86)>cd Mosquitto
C:\Program Files (x86)\Mosquitto>mosquitto_sub.exe -h localhost -t /mytopic/1
Hello MQTT test
```

발행 (Publish)

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



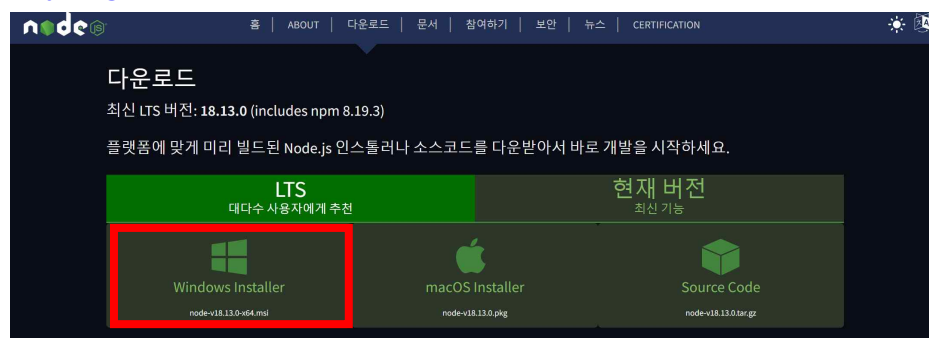
```
C:\Users\Weblee>cd "C:\Program Files (x86)"
C:\Program Files (x86)>cd Mosquitto
C:\Program Files (x86)\Mosquitto>mosquitto_pub.exe -h localhost -t /mytopic/1 -m "Hello MQTT test"
C:\Program Files (x86)\Mosquitto>
```



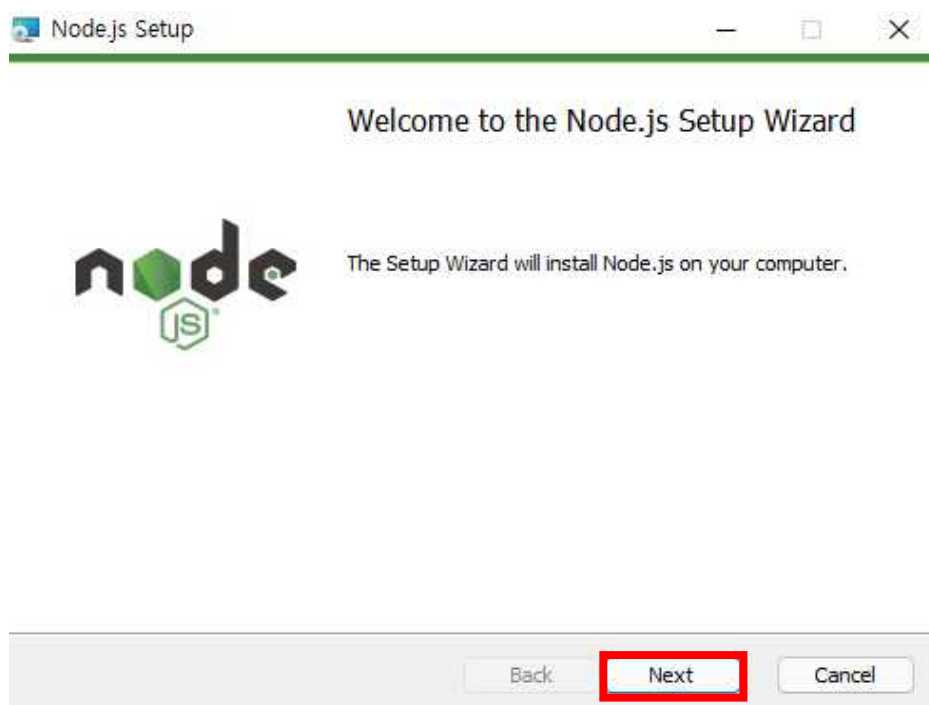
## 2.2.2. Node JS 설치

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.

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



위 링크에서 최신버전의 nodeJS 설치파일을 다운받는다.

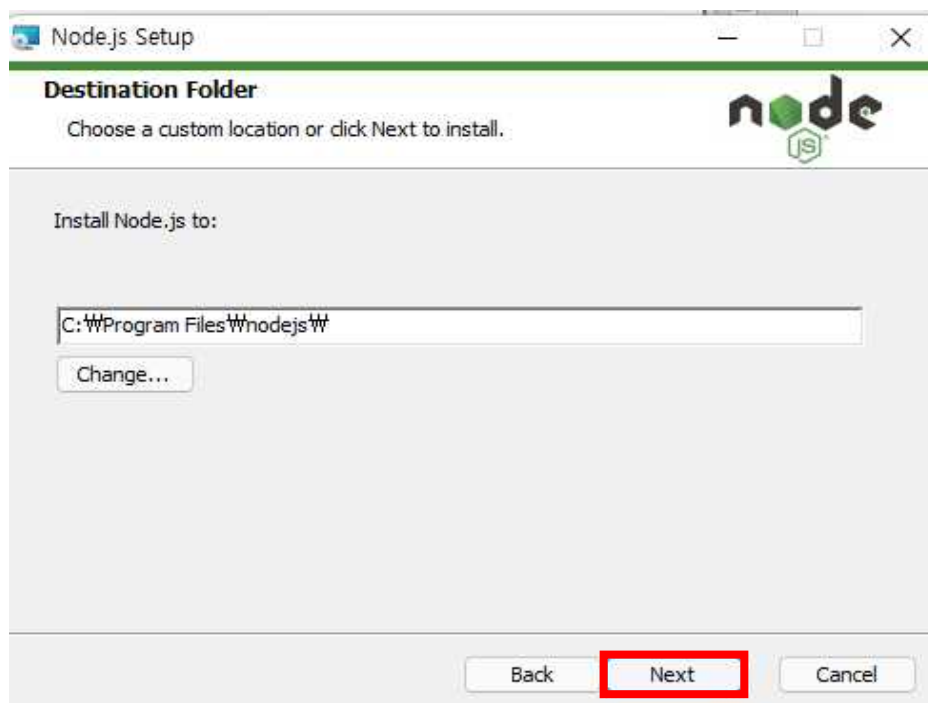


다음으로 위와 같이 다운로드한 인스톨러를 실행한다. Next를 클릭한다.

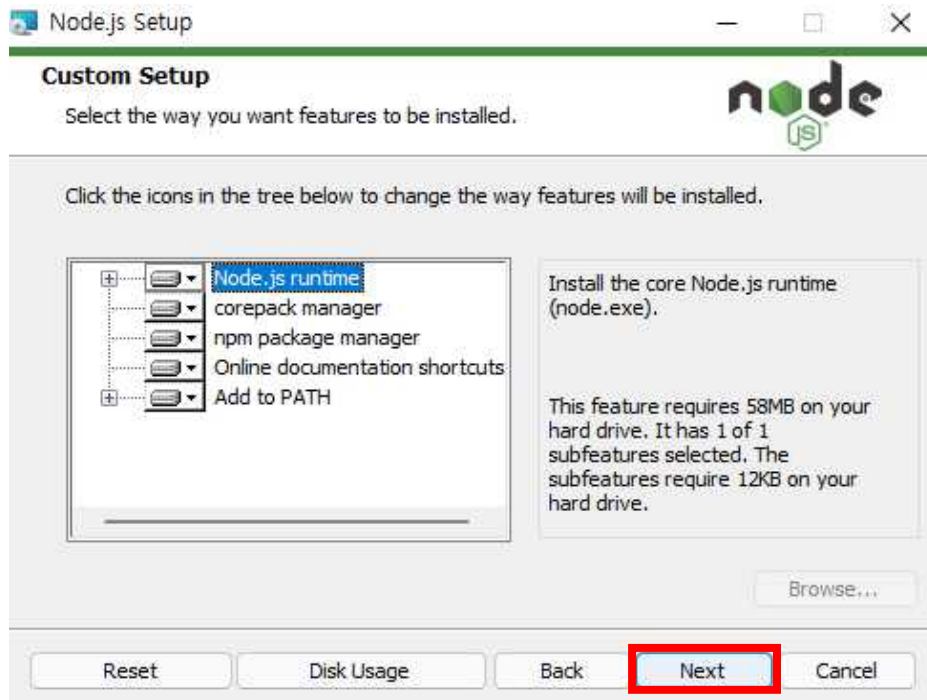




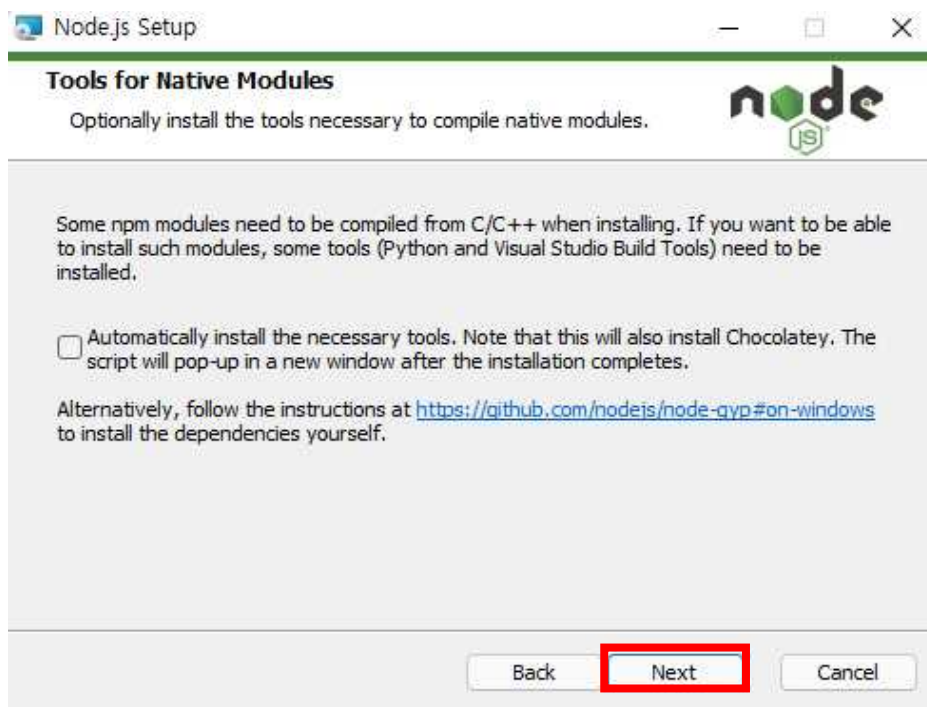
설치를 위해 라이선스에 동의한 후 Next를 클릭한다.



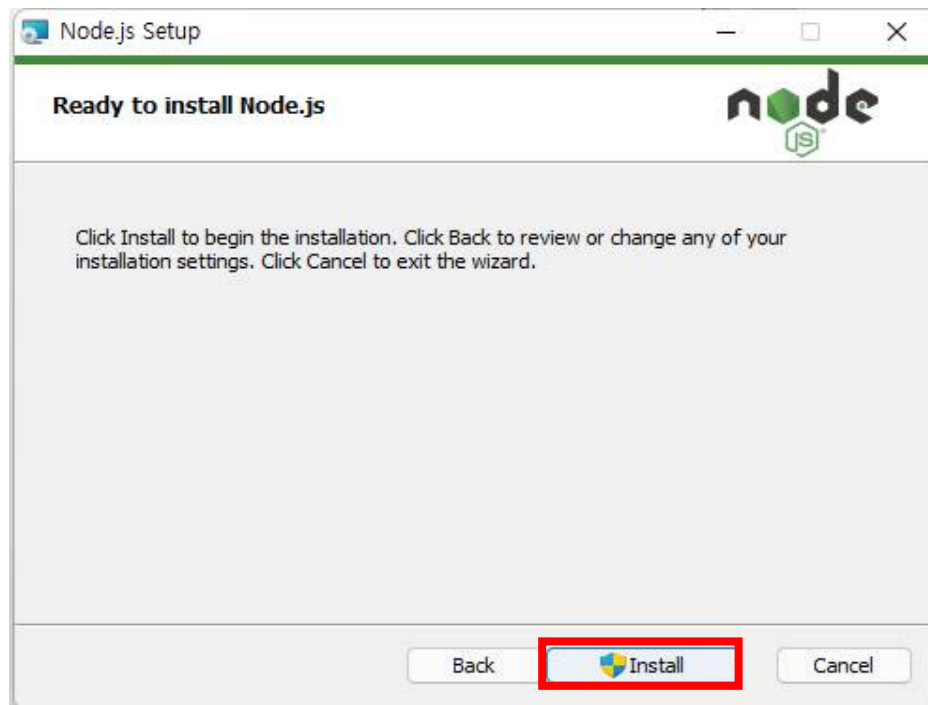
Next를 클릭한다.



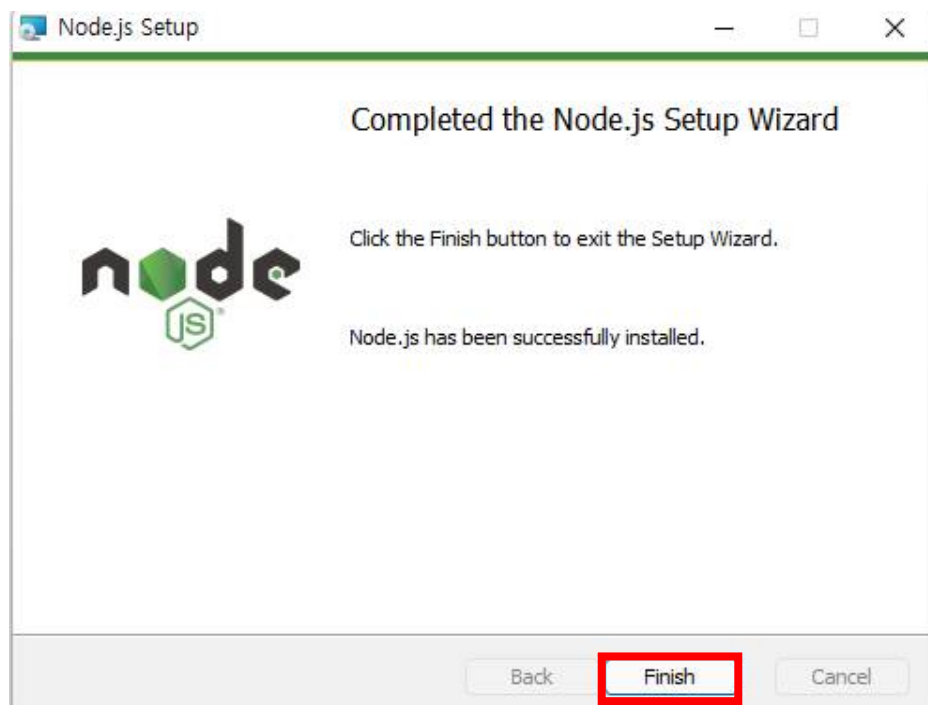
Next를 클릭한다.



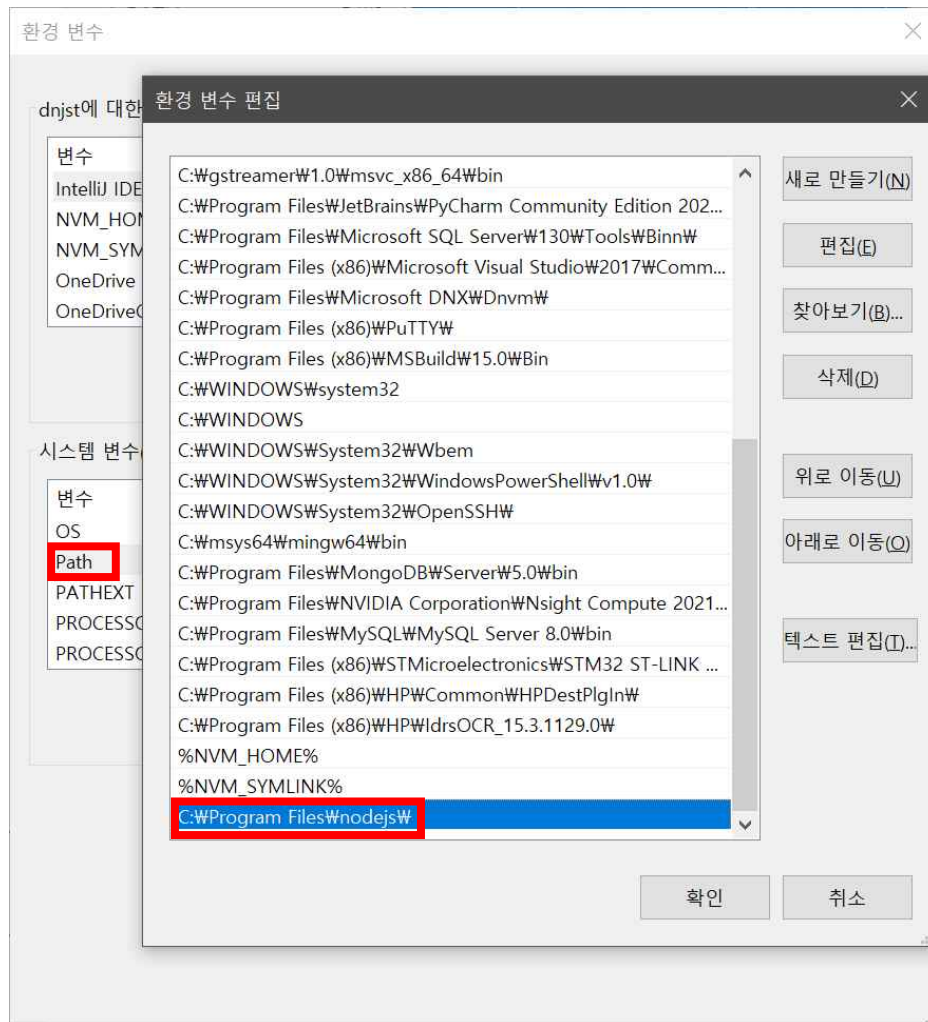
Next를 클릭한다.



설치를 위해 Install을 클릭한다.

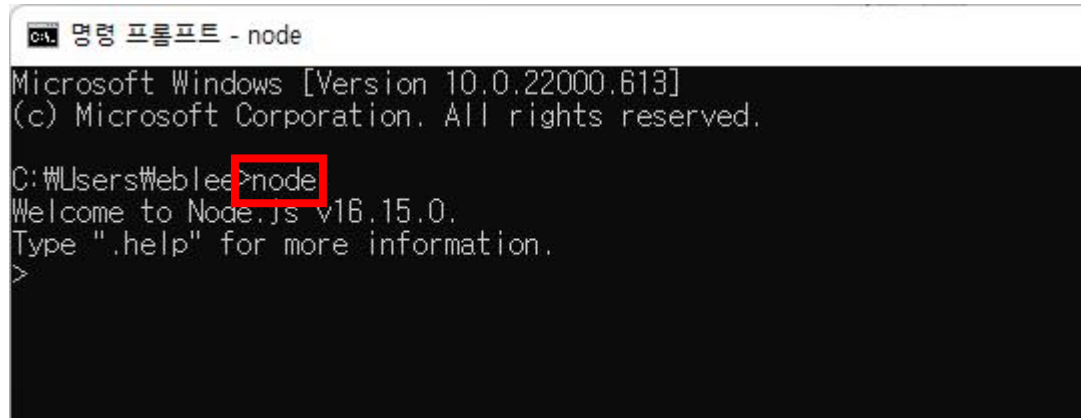


설치가 완료되면 Finish 버튼을 클릭한다.



NodeJs의 경우 설치 시, 환경 변수 설정이 자동으로 된다. 설치 후, NodeJS가 환경 변수에 추가되어 있는지 확인한다.

만약 추가되어 있지 않다면 새로 만들기를 클릭하고 위와 같이 C:\Program Files\nodejs\를 추가하고 확인 버튼을 누른다.



```
명령 프롬프트 - node
Microsoft Windows [Version 10.0.22000.613]
(c) Microsoft Corporation. All rights reserved.

C:\Users\weblee>node
Welcome to Node.js v16.15.0.
Type ".help" for more information.
>
```

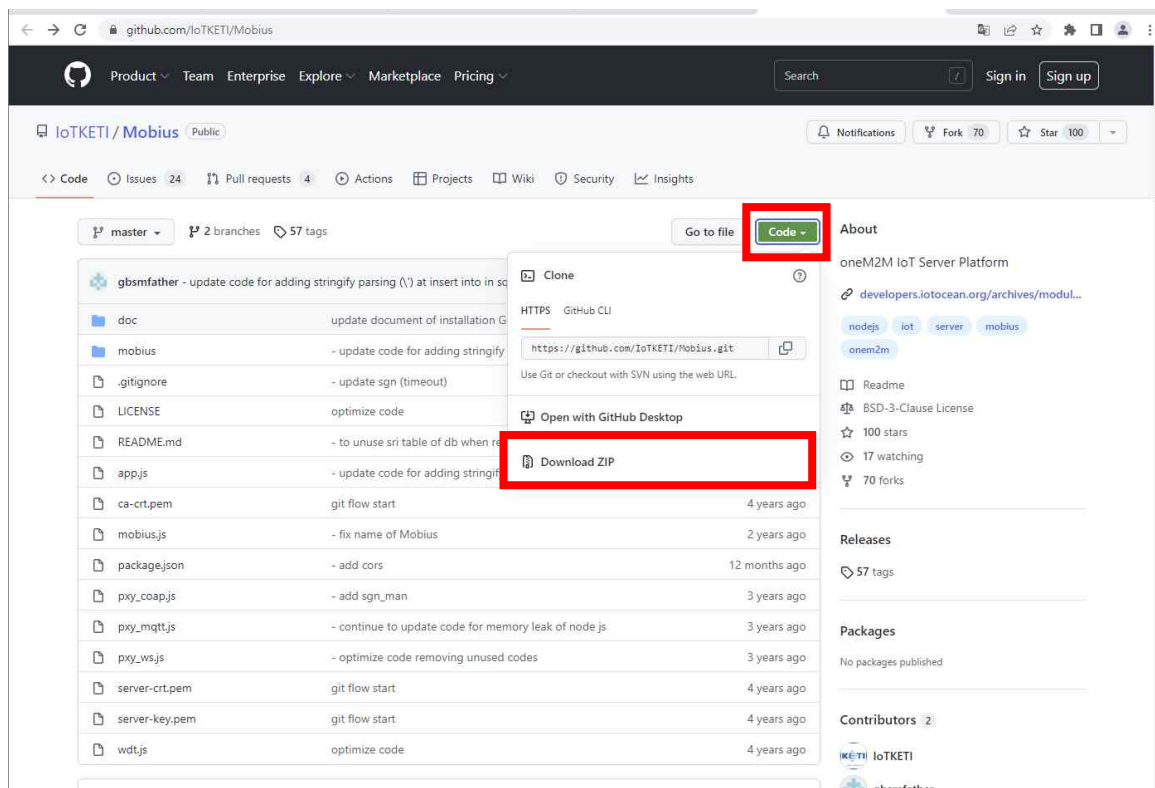
Node.js가 제대로 설치가 되었는지 확인한다. Node.js는 대화형 cli를 제공한다. 위와 같이 실행이 되면 Node.js가 제대로 설치가 된 것이다.

## 2.3. Mobius 설치

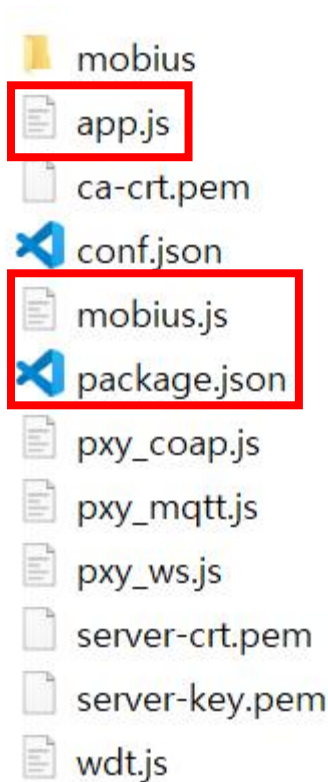
Mobius Platform은 GitHub를 통해 Mobius 서버 플랫폼 소스를 공개하고 있다. Mobius 서버 플랫폼 압축 파일을 다운로드 받은 후, 제공된 매뉴얼에 따라 Port 정보 등의 Configuration 정보를 설정한다.

### 2.3.1. 다운로드 및 모듈 설치

Mobius 설치 파일은 OCEAN Alliance에서 제공하는 GitHub (<https://github.com/loTKETI/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.js가 Mobius의 메인 코드를 실행하는 파일이다. app.js는 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.

```
명령 프롬프트
C:\Users\weblee\Desktop>cd Mobius-master
C:\Users\weblee\Desktop\Mobius-master>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 one 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 10s

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

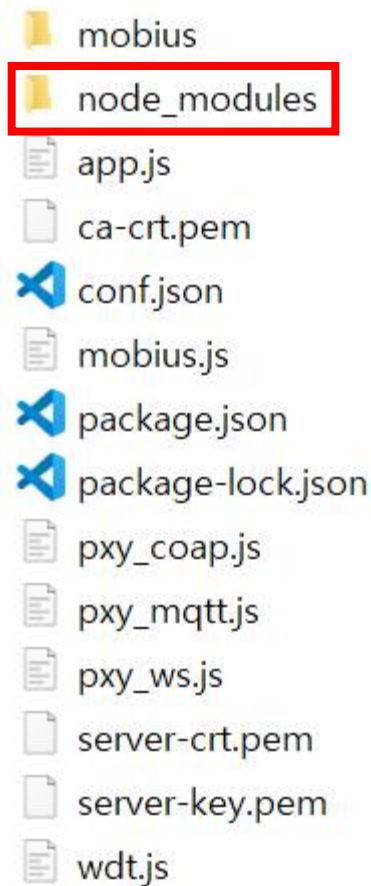
5 high severity vulnerabilities

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

Run `npm audit` for details.
npm notice
npm notice New minor version of npm available! 8.5.5 -> 8.8.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.8.0
npm notice Run `npm install -g npm@8.8.0` to update!
npm notice
C:\Users\weblee\Desktop\Mobius-master>
```

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.

```
C:\> 명령 프롬프트 - node mobius.js

C:\Users\dns00\Desktop\Mobius>node mobius.js
Production Mode
Garbage collection is not exposed
sgn_mqtt_client is connected
[db.connect] No Connection
```

이제 Mobius를 구동하기 위한 모듈 설치가 끝났다.

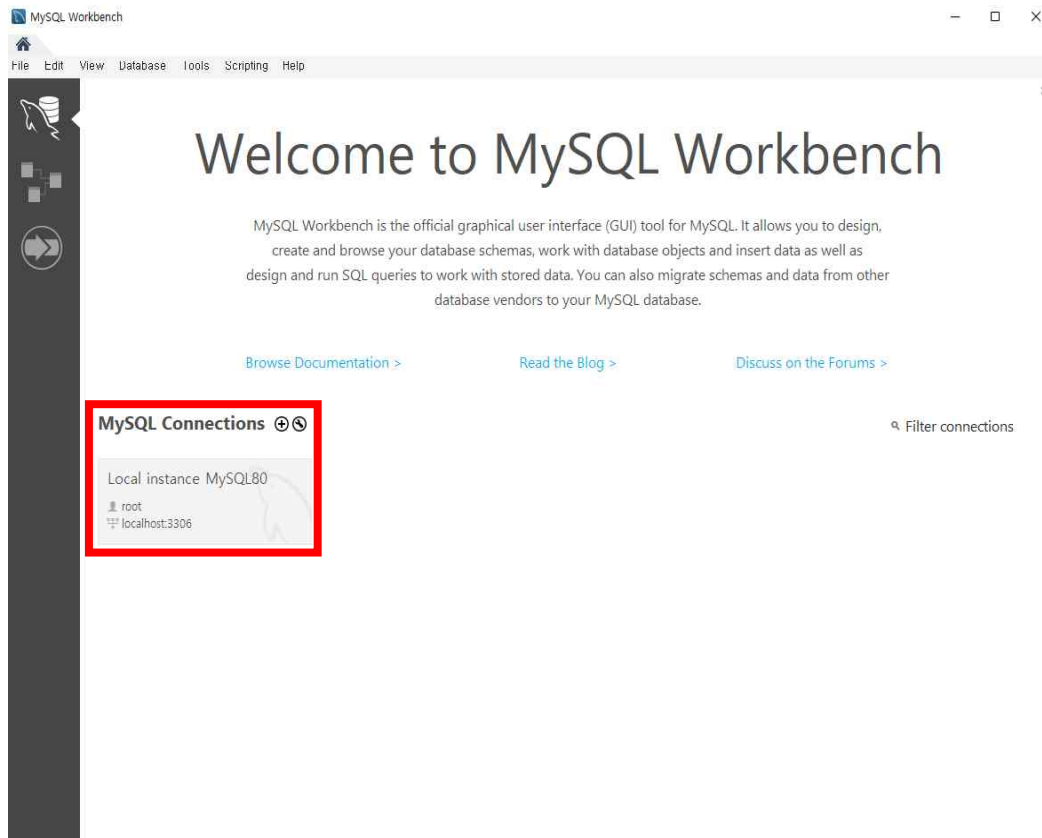
하지만 위의 그림과 같이 node mobius.js 를 실행하면 MySQL과 데이터베이스가 연동되어  
있지 않아 오류가 발생한다.

따라서 Mobius와 MySQL을 연동하는 과정이 필요하다.

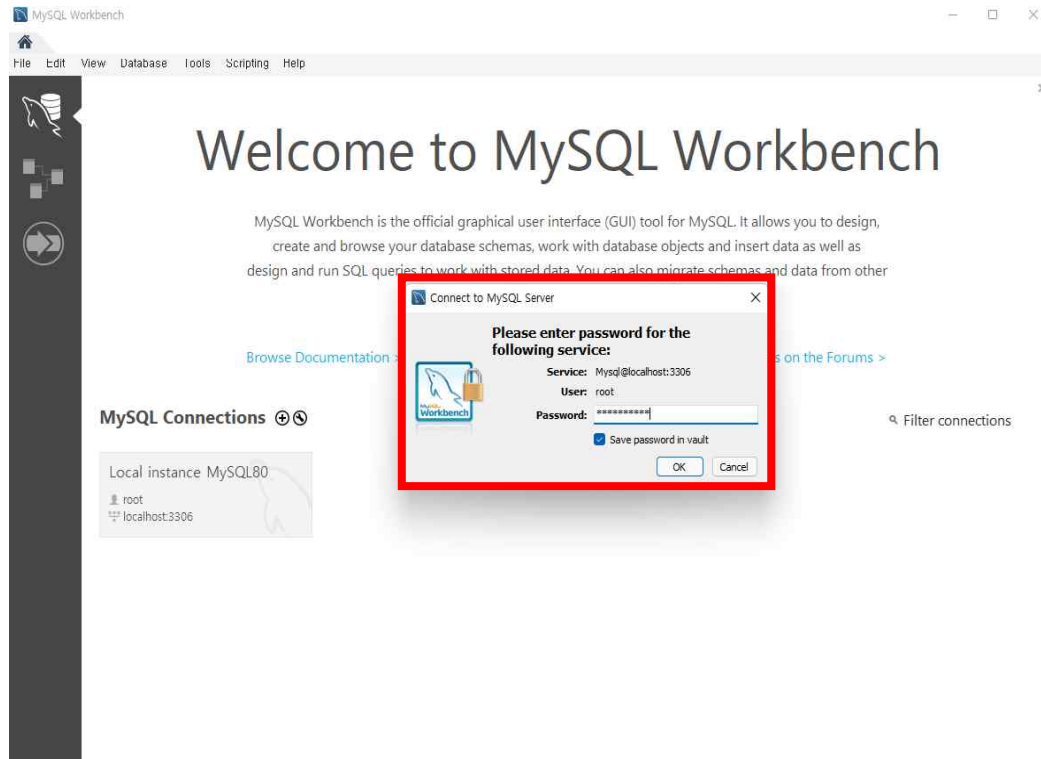


### 2.3.2. mobiusdb 데이터베이스 생성

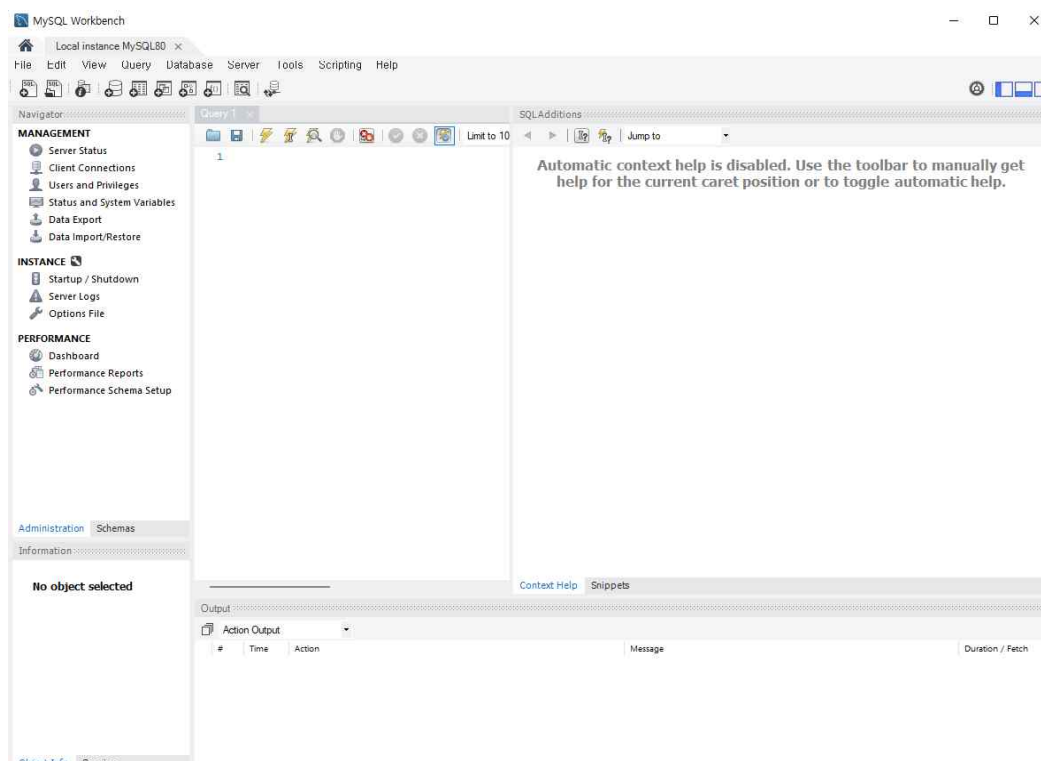
위의 Mobius 구동 개발환경에서 MySQL을 설치했으므로 MySQL Workbench가 설치되어 있을 것이다.



MySQL Workbench를 실행하고 Connections를 선택한다.

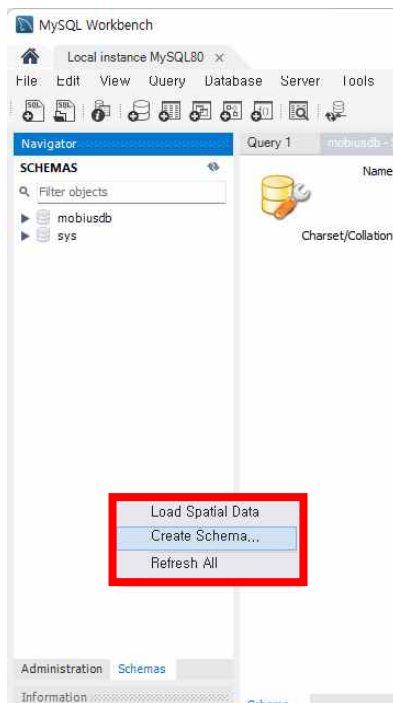


MySQL을 설치할 때 설정한 암호를 입력한다.

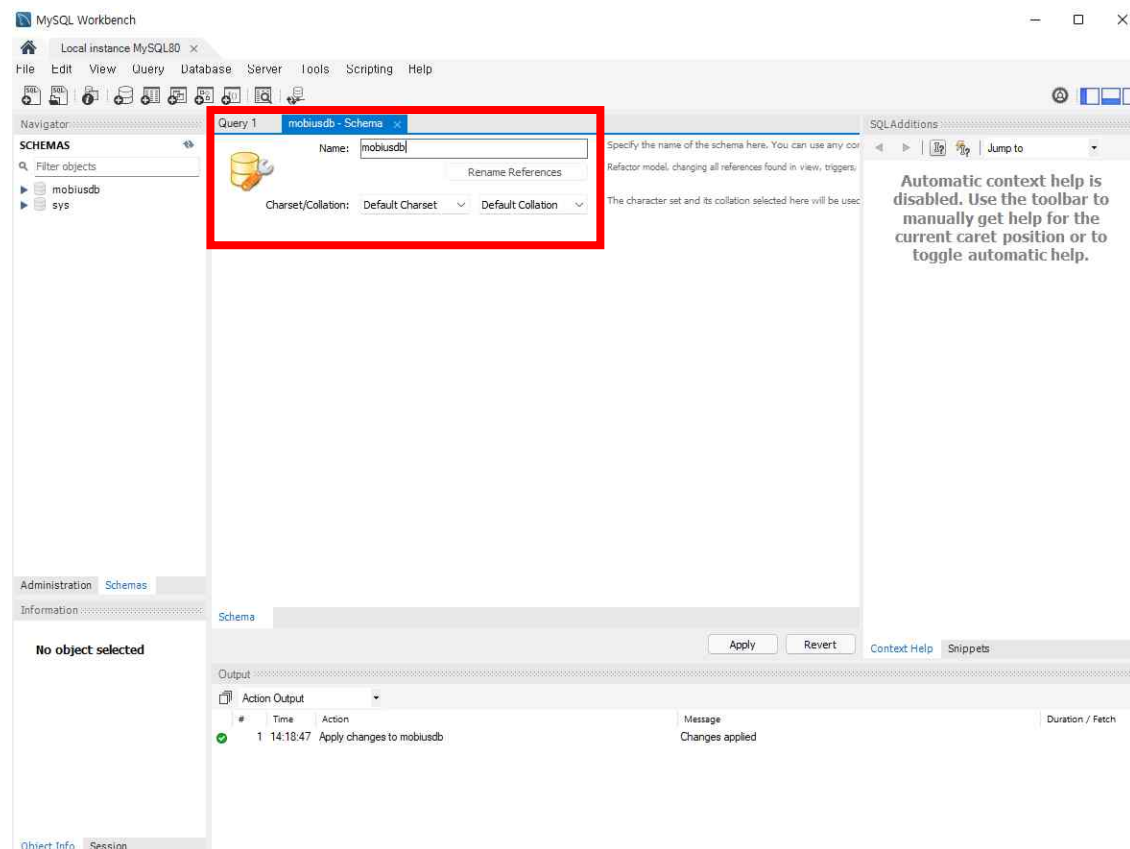


Workbench는 GUI환경에서 MySQL을 다룰 수 있는 프로그램으로 SQL 쿼리를 날릴 수 있는 것은 물론 데이터베이스, 테이블 생성, Import, Export 등 다양한 MySQL 관리 작업을 할 수 있다.

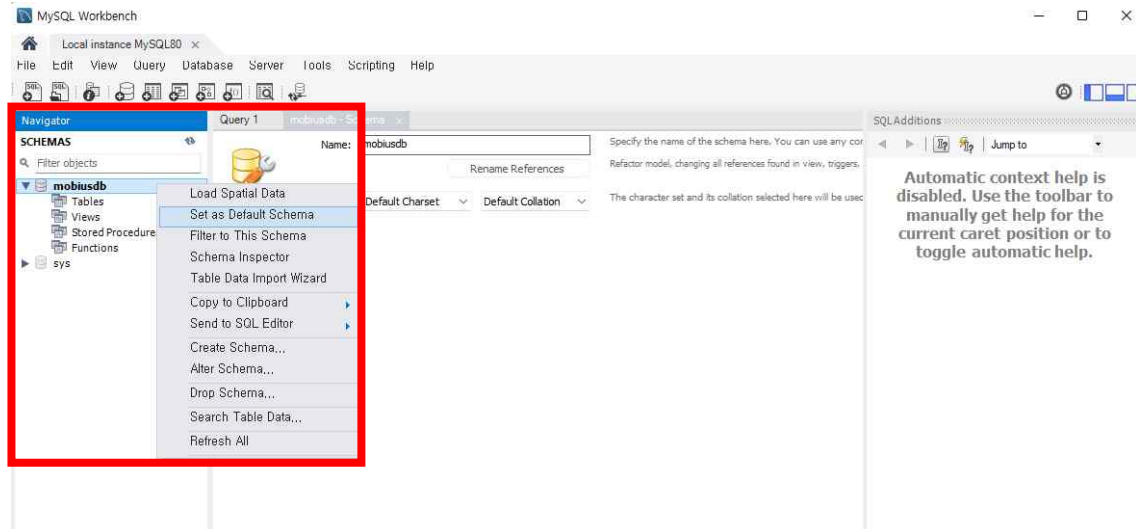
## - 데이터베이스 생성



위의 그림처럼 SCHEMAS 빈 곳에서 마우스 오른쪽 버튼을 클릭하면 뜨는 메뉴에서 **"Create Schema..."**를 선택한다.



스키마 생성 창이 뜨면 데이터베이스 이름을 **"mobiusdb"** 입력하고 Apply 버튼을 누르면 데이터베이스가 생성된다.

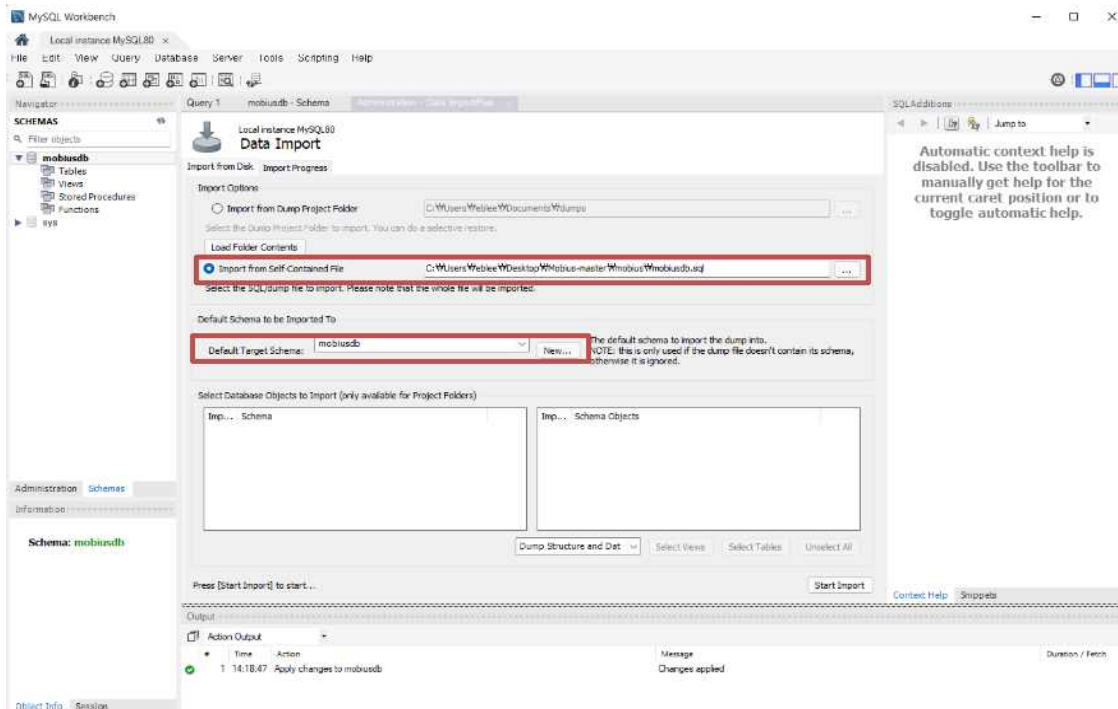


"mobiusdb" 데이터가 생성이 되면 위의 그림과 같이 왼쪽 창에 데이터베이스가 생성됨을 확인할 수 있고 마우스 오른쪽 버튼을 눌러 **Set as Default Schema**로 설정을 해 준다.

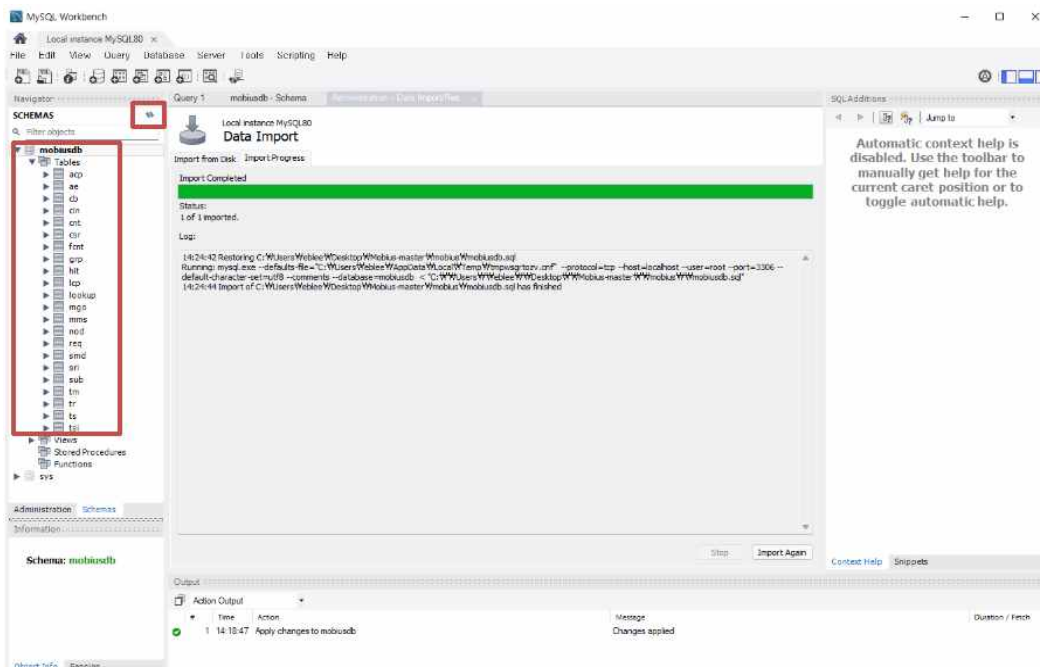
- mobiusdb로 테이블 가져오기

acp.js  
ae.js  
asn.js  
cb.js  
cin.js  
cnt.js  
cnt\_man.js  
csr.js  
db\_action.js  
fcnt.js  
fopt.js  
grp.js  
lcp.js  
mgo.js  
mms.js  
mn.js  
**mobiusdb.sql**  
nod.js  
req.js  
resource.js  
responder.js  
security.js  
sgn.js  
sgn\_man.js  
smd.js  
sql\_action.js  
sub.js  
tm.js  
tr.js  
ts.js  
ts\_agent.js  
tsi.js

Import할 Mobius DB 테이블에 대한 sql 파일은 앞서  
다운로드 한 Mobius가 설치된 폴더 내의 mobius 폴더에  
위치해 있다.



mobiusdb로 테이블을 가져오기 위해 Workbench의 Server 메뉴에서 Data Import를 선택한다. Data Import 창에서 Import from Self-Contained File을 선택하고 위의 그림에 있는 mobiusdb.sql 파일을 선택한다. 이후 Default Target Schema로 앞서 만든 mobiusdb를 선택하고 Start Import를 누르면 다음 그림과 같이 Mobius에서 사용되는 테이블이 생성된다.



Data Import를 수행한 후에는 테이블이 보이지 않으면 위에 있는 새로고침 아이콘을 클릭해야 Import 된 테이블들을 볼 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.

```
C: > Users > eblee > Desktop > Mobius-master > {} conf.json > ...  
1  {  
2    "csebaseport": "7579",  
3    "dbpass": "mysqladmin"  
4  }
```

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.

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

```

선택 명령 프롬프트 - node mobius.js
C:\Users\Wdms00\Desktop\Mobius>node mobius.js
Production Mode
Garbage collection is not exposed
sgn_mqtt_client is connected
CPU Count: 8
select_ri_lookup /Mobius: 1.919ms
update_cb_poa_csi /Mobius: 0.642ms
...
Production Mode
Garbage collection is not exposed
sgn_mqtt_client is connected
Production Mode
Garbage collection is not exposed
Production Mode
mobius server (192.168.50.163) running at 7579 port
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
select_ri_lookup /Mobius: 1.474ms
update_cb_poa_csi /Mobius: 0.603ms
...
mobius server (192.168.50.163) running at 7579 port
mobius server (192.168.50.163) running at 7579 port
mobius server (192.168.50.163) running at 7579 port
Production Mode
Garbage collection is not exposed
select_ri_lookup /Mobius: 1.375ms
select_ri_lookup /Mobius: 1.253ms
select_ri_lookup /Mobius: 1.244ms
update_cb_poa_csi /Mobius: 0.497ms
update_cb_poa_csi /Mobius: 0.412ms
...
update_cb_poa_csi /Mobius: 0.426ms
...
sgn_mqtt_client is connected
Production Mode
Garbage collection is not exposed
mobius server (192.168.50.163) running at 7579 port
Production Mode
Garbage collection is not exposed
Garbage collection is not exposed
sgn_mqtt_client is connected
select_ri_lookup /Mobius: 1.438ms
update_cb_poa_csi /Mobius: 0.509ms
...
sgn_mqtt_client is connected
sgn_mqtt_client is connected
mobius server (192.168.50.163) running at 7579 port
mobius server (192.168.50.163) running at 7579 port
mobius server (192.168.50.163) running at 7579 port
select_ri_lookup /Mobius: 1.265ms
update_cb_poa_csi /Mobius: 0.403ms
...
select_ri_lookup /Mobius: 1.305ms
select_ri_lookup /Mobius: 1.174ms
update_cb_poa_csi /Mobius: 0.381ms
...
update_cb_poa_csi /Mobius: 0.376ms
...
oxyws server (192.168.50.163) running at 7577 port
GET : /Mobius
GET : /Mobius
get_resource_from_url (o2c01SP11) - /Mobius: 3.354ms
get_resource_from_url (ZerbyZrXY) - /Mobius: 3.411ms
["m2m:cb":{"pi":null,"ri":"5-20210929095706523482","ty":5,"et":"20210929T095706","rn":"Mobius","lt":"20210929T095706","lbi":["Mobius"],"cst":1,"csi":"/Mobius2","srt":["1,2,3,4,5,9,10,13,14,16,17,23"],"poa":["http://192.168.50.163:7579"],"srv":["1","2","2a"]}]
["m2m:cb":{"pi":null,"ri":"5-20210929095706523482","ty":5,"et":"20210929T095706","rn":"Mobius","lt":"20210929T095706","lbi":["Mobius"],"cst":1,"csi":"/Mobius2","srt":["1,2,3,4,5,9,10,13,14,16,17,23"],"poa":["http://192.168.50.163:7579"],"srv":["1","2","2a"]}]
express deprecated res.status("200"): use res.status(200) instead mobius#responder.js:1850:22
express deprecated res.status("200"): use res.status(200) instead mobius#responder.js:1850:22
...
subscribe req_topic as /oneM2M/req/+ /Mobius2/+
subscribe req_topic as /oneM2M/req req/+ /Mobius2/+
ts_missing agent server (192.168.50.163) running at 7582 port
GET : /Mobius?fu=1&ty=29
get_resource_from_url (8tVtnUDVX) - /Mobius?fu=1&ty=29: 3.798ms
search_parents_lookup /Mobius: 114.641ms
search_lookup (dnAVb8MtZE): 4.987ms
express deprecated res.status("200"): use res.status(200) instead mobius#responder.js:2070:22
...
init_TS - 4004

```

### 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

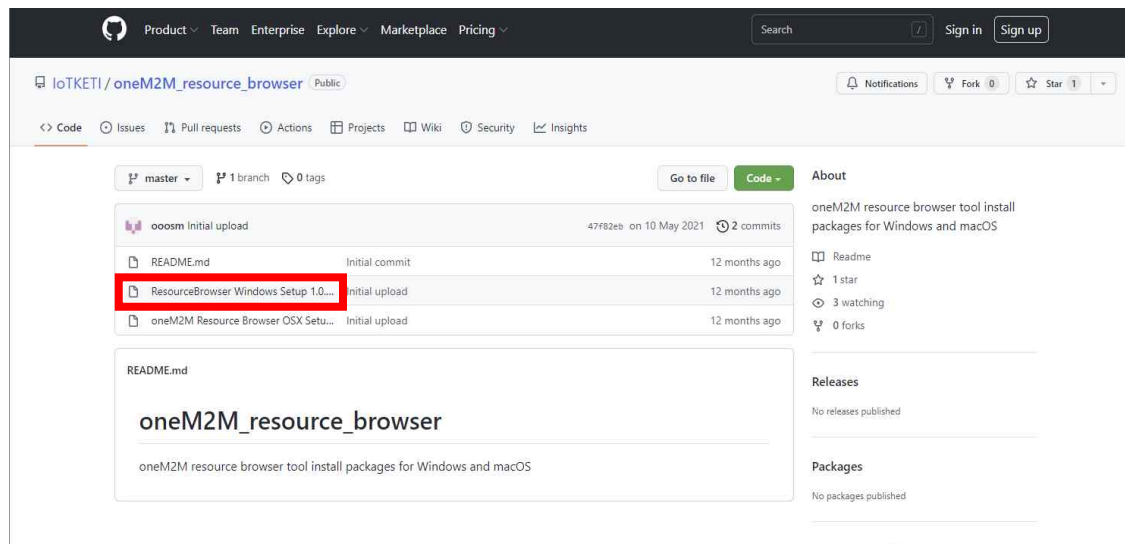
### 3.2.1. 리소스뷰어(oneM2M Browser) 활용

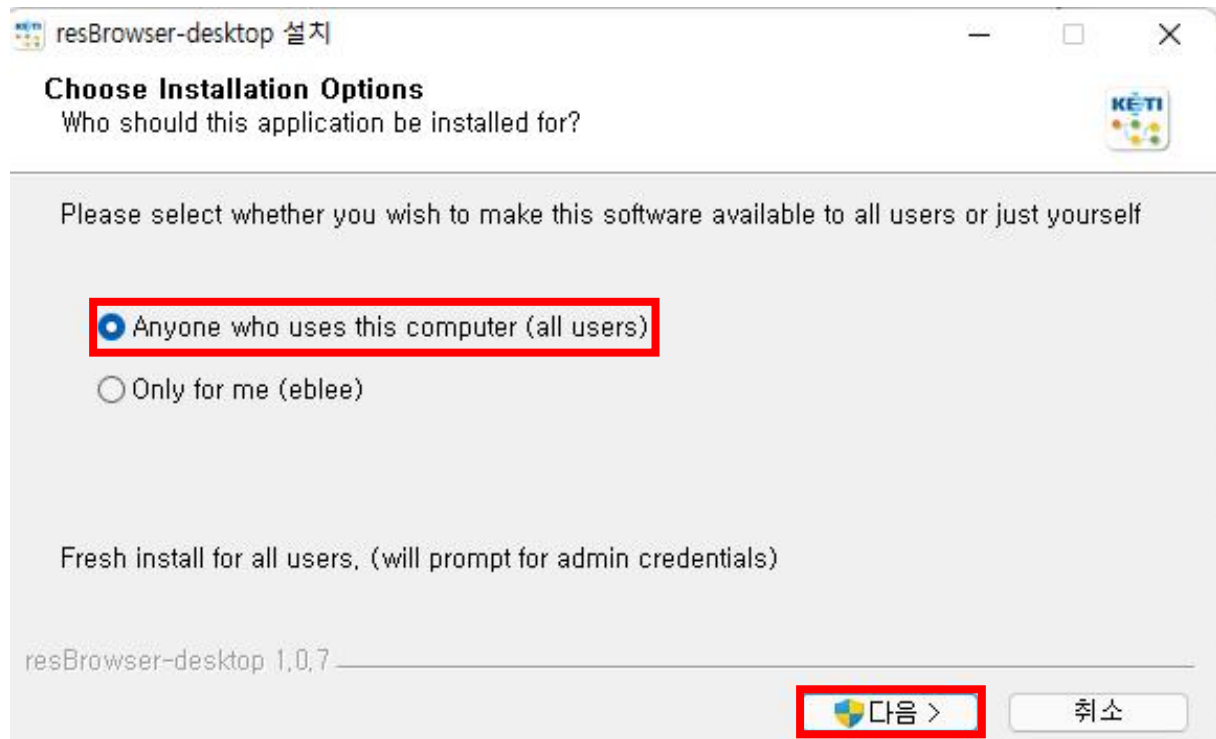
Mobius와 nCube를 통해 만들어진 디바이스가 제대로 동작하는지 테스트하기 위해 웹으로 제공하는 리소스뷰어를 사용할 수 있다. 리소스뷰어는 윈도우 파일 탐색기와 비슷하게 트리 구조의 리소스를 보여주고 리소스 생성, 삭제의 간단한 동작을 수행할 수 있다.

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.

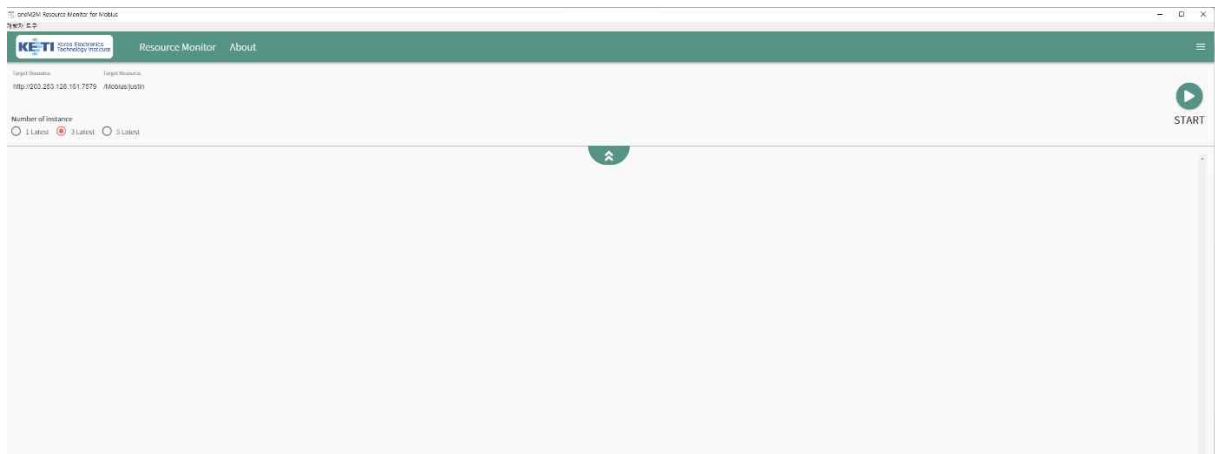




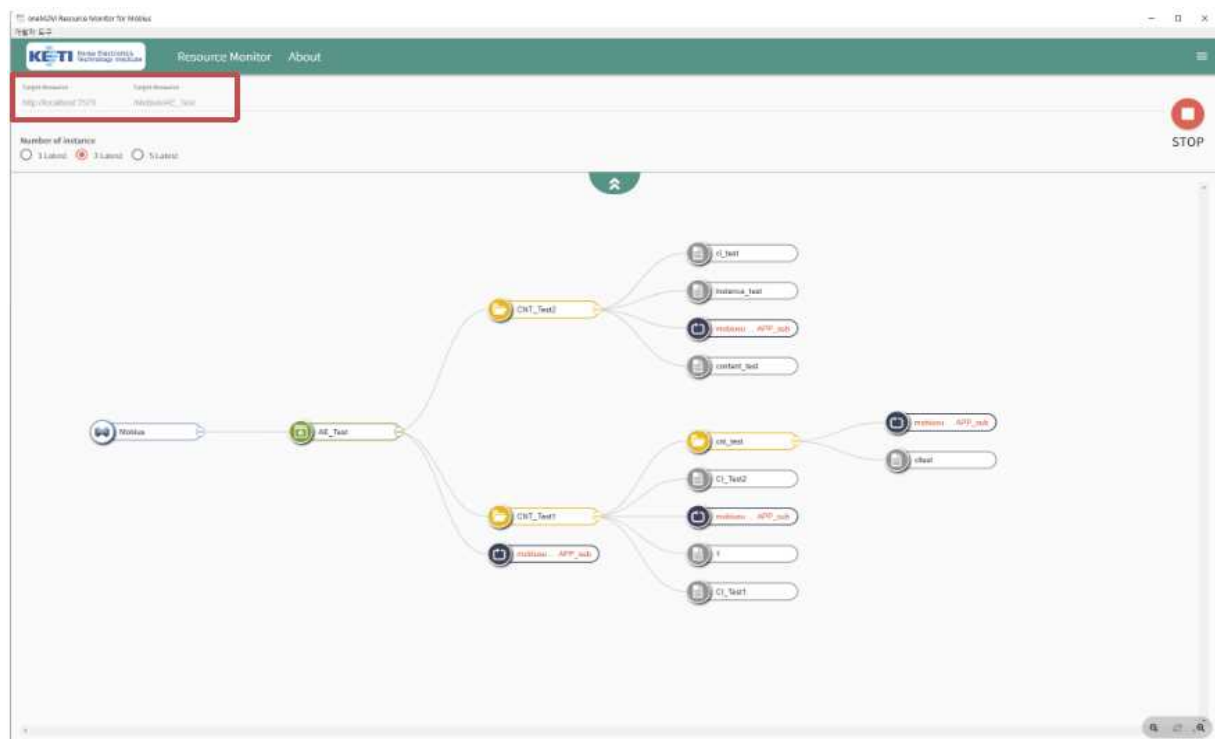
리소스뷰어 설치를 위해 다음을 클릭한다.



설치가 완료되면 마침을 클릭한다.



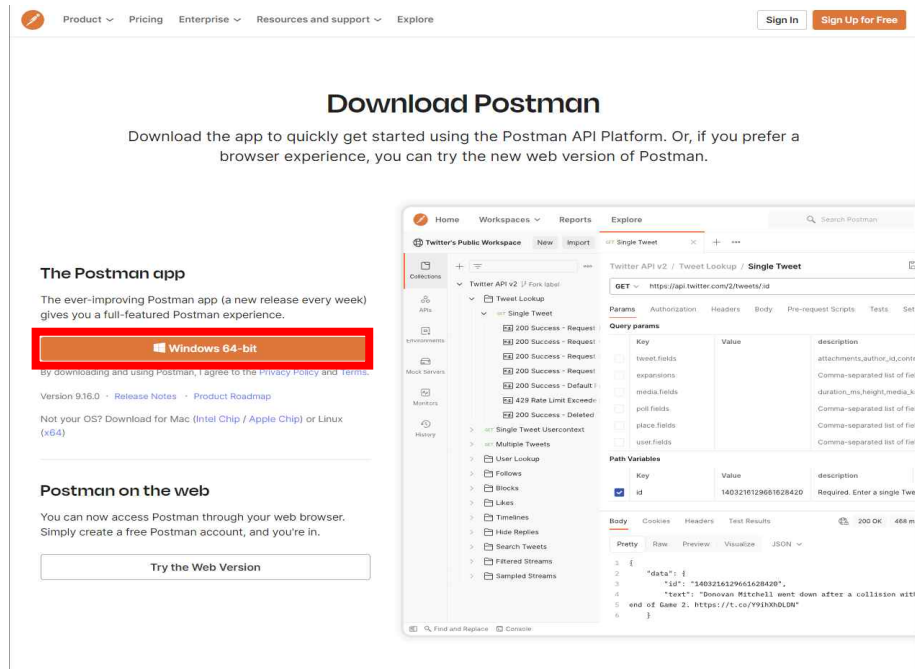
리소스뷰어 설치가 완료되면 위와 같이 실행된다.



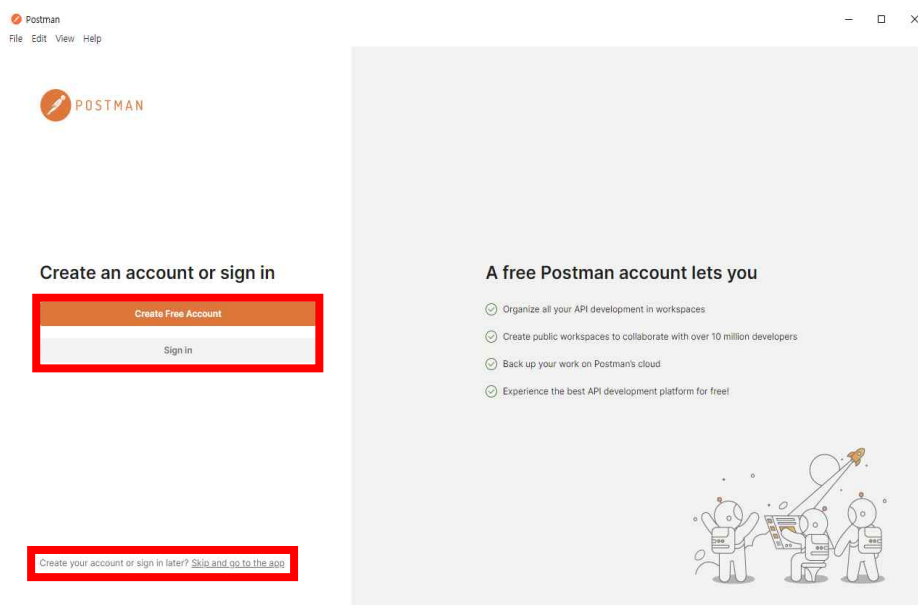
아래와 같이 리소스뷰어가 보이면 표신된 부분에 Mobius에 생성된 리소스의 url을 입력하고 Strat 버튼을 누르면 해당 url의 리소스 구조를 볼 수 있다. url 부분의 처음은 콤보 박스로 되어 있고 이전 Mobius와 Mobius release 2를 선택할 수 있다. 이후 부분은 직접 입력해야 하고 여기에 생성한 AE의 name을 입력하면 된다.

### 3.2.2. Postman 활용

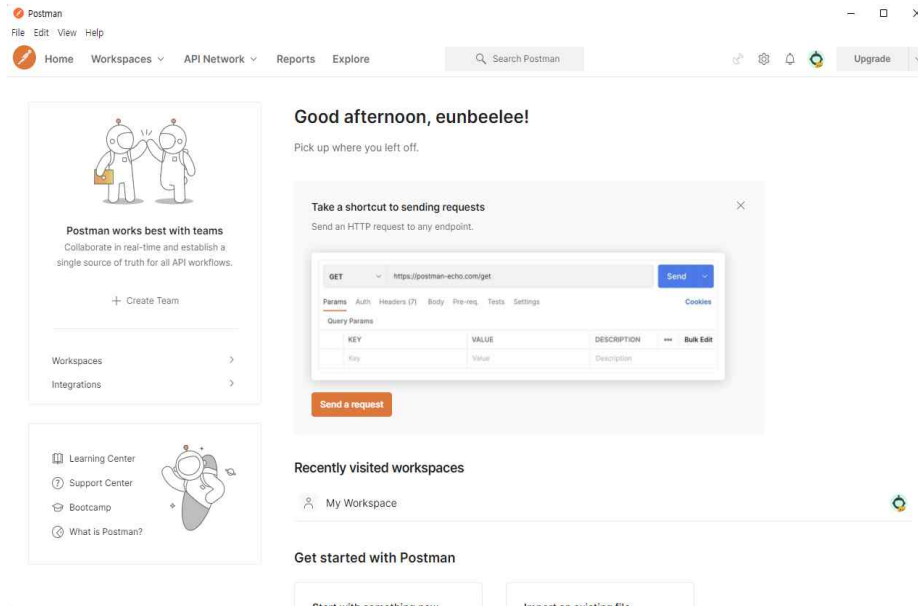
Postman을 활용하여 직접 구축한 Mobius를 동작을 테스트 할 수있다.



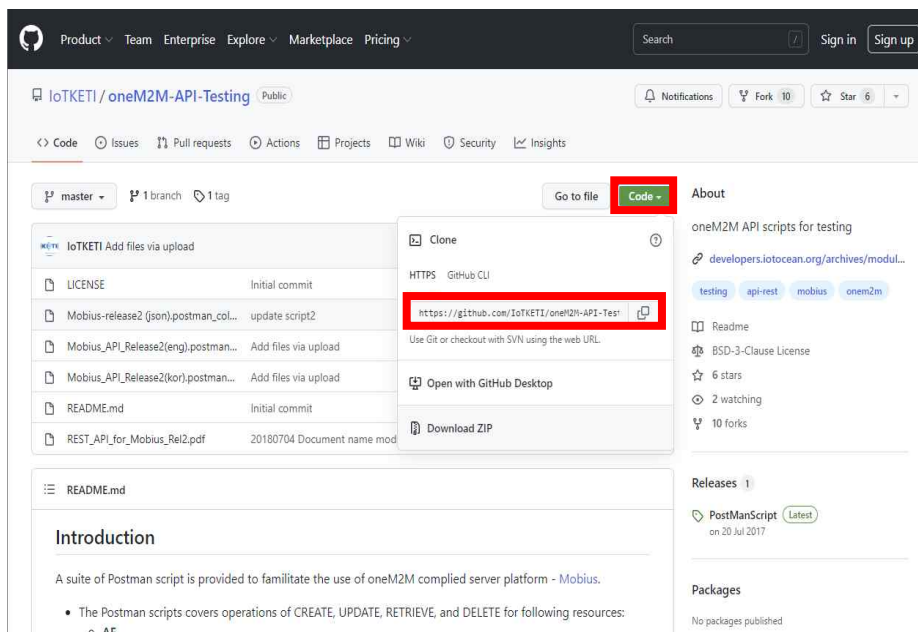
<https://www.postman.com/downloads/> 해당 사이트에 접속하여 postman을 다운받을 수 있다.



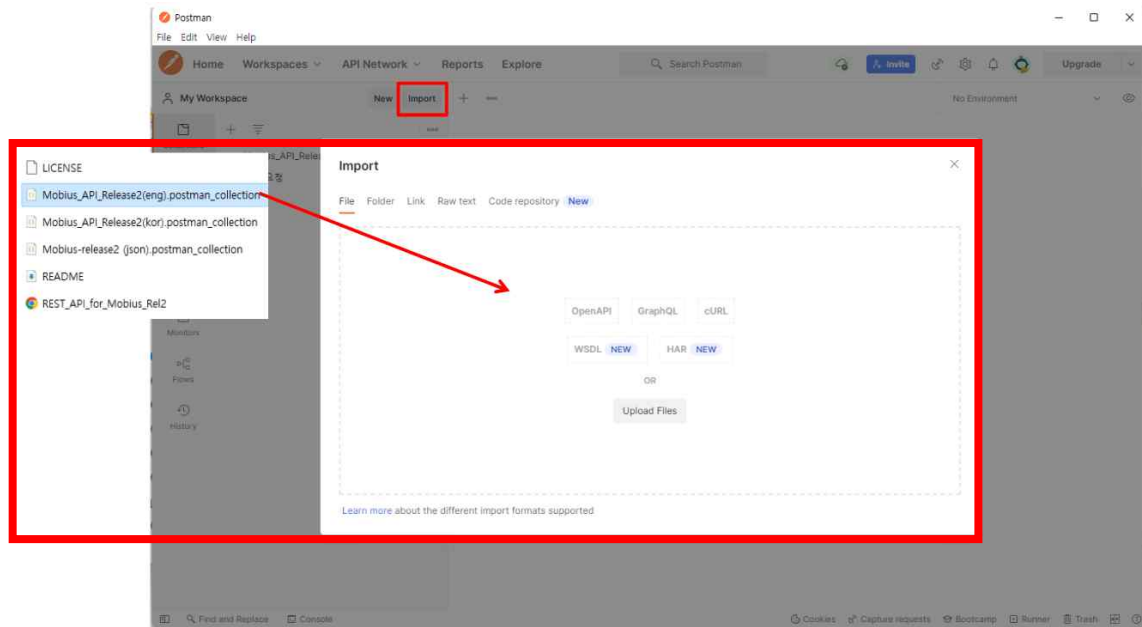
Postman을 설치하면 회원가입 후 로그인한다.  
로그인을 원하지 않으면 아래에 Skip 하면된다.



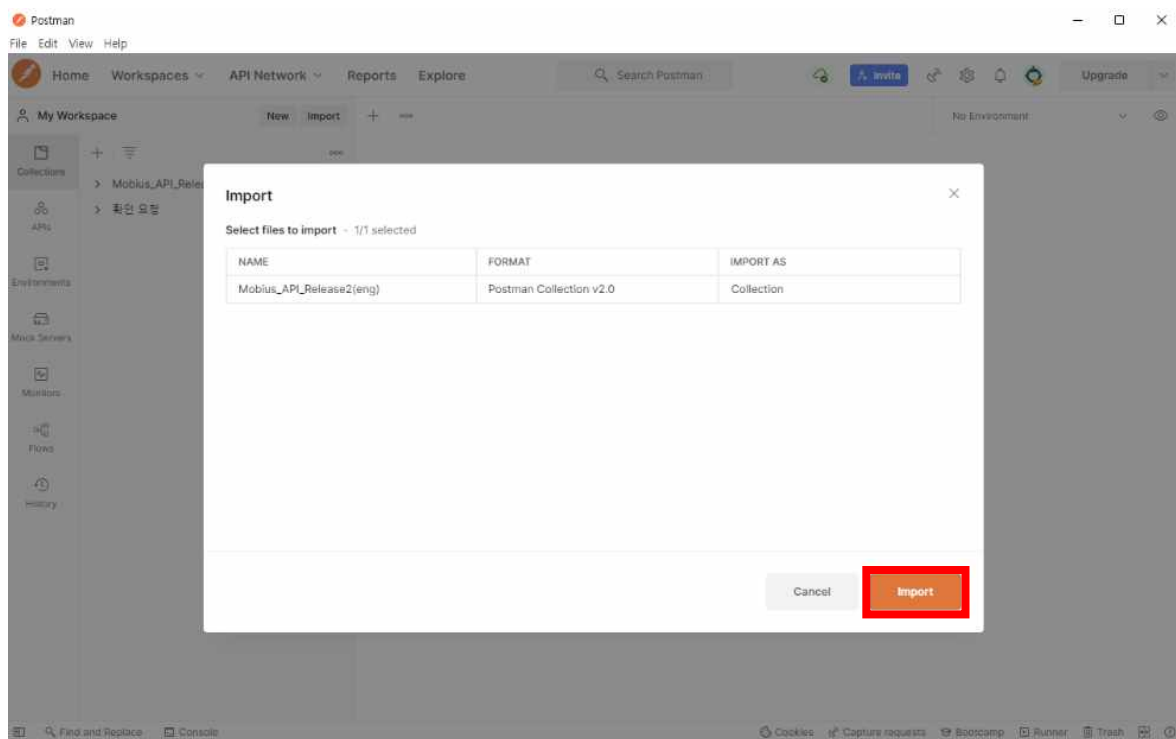
Postman 설치 후 로그인하면 위와 같이 실행된다.



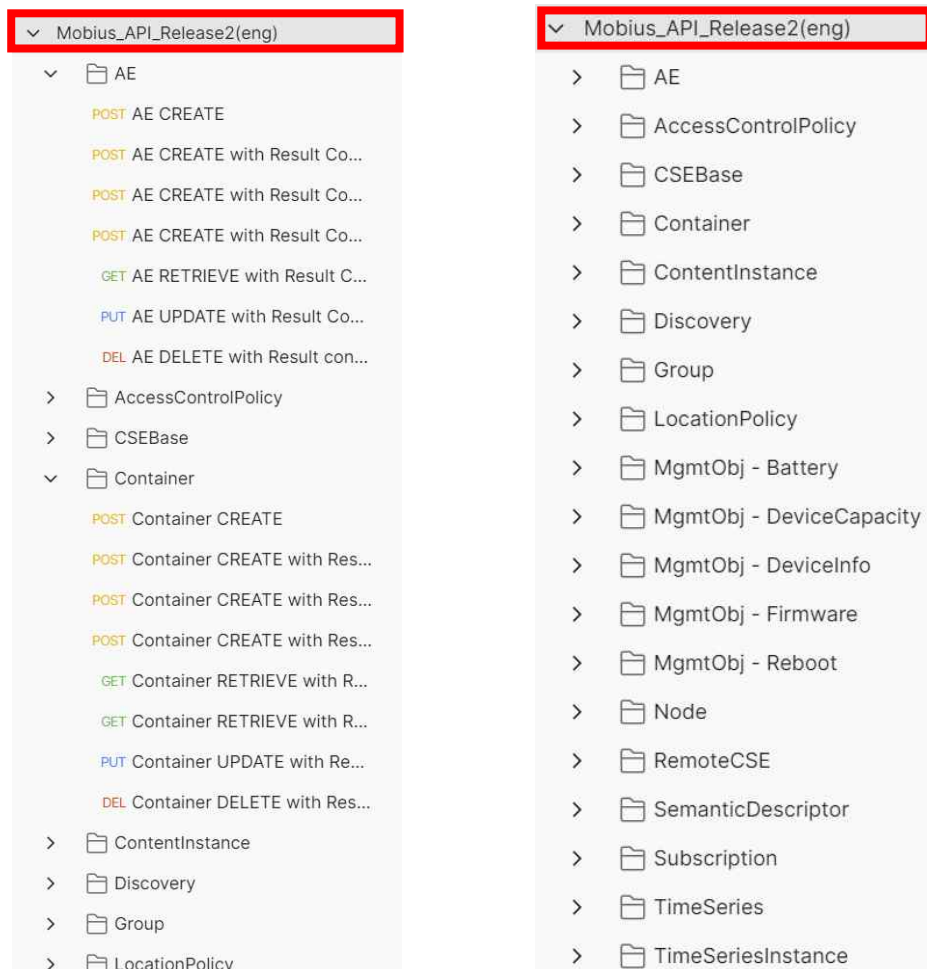
Mobius 테스트를 위해 OCEAN에서 미리만들어 둔 스크립트를 다운받는다.



Import 하는 방법은 그림과 같이 My Workspace 옆에 Import 버튼을 선택한다.  
Import를 위한 창이 열리면 OCEAN에서 다운로드 받은 스크립트 파일을 드래그한다.



Import 버튼을 클릭하여 import를 완료한다.

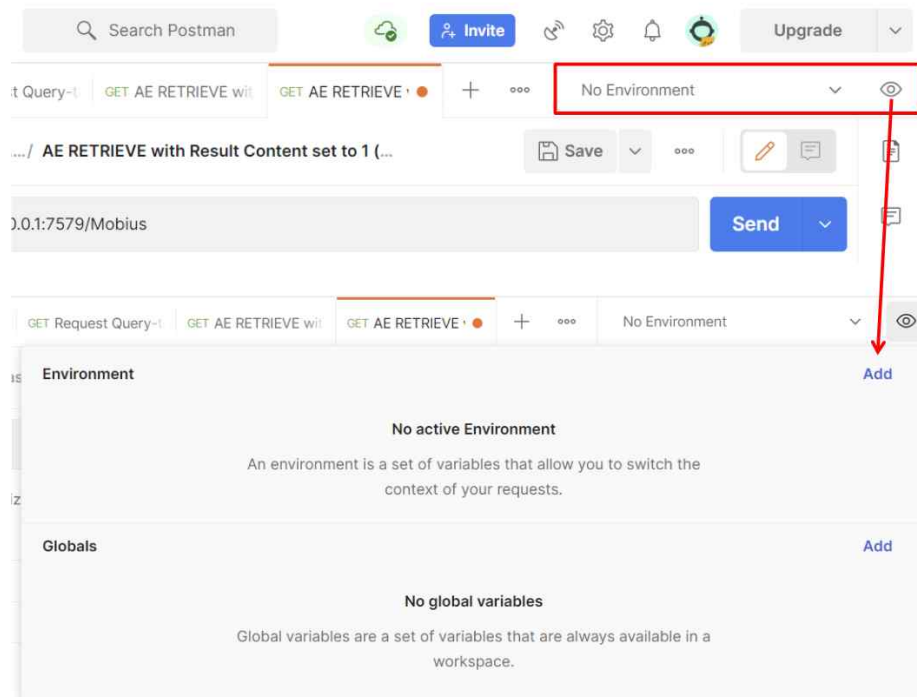


Import가 성공하면 위의 그림처럼 Mobius\_API\_Release2(eng)이라는 컬렉션 탭이 생성이 될 것이다. 이 테스트 스크립트는 현재 지원되는 리소스의 CRUD를 할 수 있도록 정리를 해 둔 것이고 xml 데이터 포맷으로 테스트 할 수 있도록 설정되어 있다.



테스트를 하기 위해서는 Mobius가 구동중인 서버의 IP를 입력해야 한다.

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



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

Environment 탭에서 Add를 선택한다.

New Environment					Fork 0		Save
	VARIABLE	TYPE ①	INITIAL VALUE ②	CURRENT VALUE ③	...	Pe	
<input checked="" type="checkbox"/>	mp_url	default	http://127.0.0.1:7579	http://127.0.0.1:7579			
<input checked="" type="checkbox"/>	cb	default	Mobius	Mobius			

설정 창이 나오면 아래와 같이 {{mp\_url}}, {{cb}} 값을 입력한다.

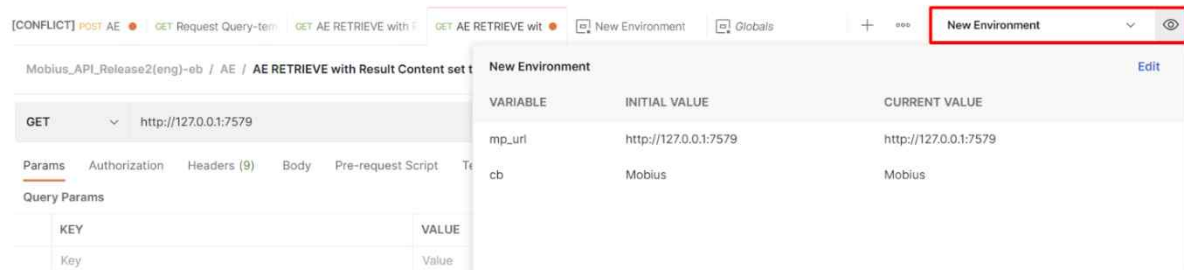
① VARIABLE 이름에 mp\_url, cb를 입력한다.

② mp\_url의 INITIAL VALUE에 http://127.0.0.1:7579를 입력한다. CURRENT VALUE는 자동으로 입력된다.

③ cb의 INITIAL VALUE에 Mobius를 입력한다. CURRENT VALUE는 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.