



Mobius Release 2

Software Version: 2.5.8

Installation Guide v2.0.0

Document Release Date: May 2022

Software Release Date: Feb 2022



Revision History

Version	Date of change	Revision History	Writer	Reviewer
		draft		
2.0.0	2018-02-11	-	Yun-gi Park	Seung-myeong Jeong/ Ill-yeop Ahn
	2022-05-09	- Additional content based on SW version update	Eun-bee Lee/ Hyeon-seo Son	Seung-myeong Jeong/ Ill-yeop Ahn

hyeonseo0128@keti.re.kr / Hyeon-seo Son KETI Researcher
eunbeelee@keti.re.kr / Eun-bee Lee KETI Researcher
monto8790@gmail.com / Yun-gi Park KETI Researcher
sm.jeong@keti.re.kr / Seung-myeong Jeong KETI Senior Researcher
iyahn@keti.re.kr / Ill-yeop Ahn KETI chief Researcher

Copyright and Disclaimer

This manual contains inaccuracies or typographical errors, and the author is not responsible for them.

The contents of this manual may be changed or added periodically, and the corresponding changes will be added sequentially to this manual in the form of "New Edition". You may change or modify the products and programs described in this manual at any time without prior notice.

The source code for [the Mobius server platform](#) is distributed according to the licensing policy below.

- OCEAN-based open source code is distributed based on the 3-clause BSD-style license. Under the 3-clause BSD-style license, OCEAN-based open source code is freely available for commercial or non-commercial systems while maintaining copyright headers in source code files.
- OCEAN-based open source licenses do not require sharing the developed source code with others. The source code developed using OCEAN-based open source belongs to the developer and is not obligated to share it with anyone.
- Anyone can contribute to OCEAN's open source development. However, if the developed source code wants to contribute to OCEAN, this code is required to comply with OCEAN's licensing policy.

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.

DISCLAIMER: 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 Introduction.....	5
1.1.	What is Mobius?.....	5
1.2.	Mobius Platform.....	6
1.2.1.	Mobius Server Platform Outline.....	6
1.2.2.	Mobius Server Platform Key Features.....	7
1.2.3.	Mobius Server Platform Configuration.....	7
1.2.4.	Mobius Server Platform S/W Architecture.....	8
1.2.5.	Mobius Server Directory Configuration.....	9
2.	Mobius Server Platform Installation.....	13
2.1.	Installation Outline.....	13
2.2.	Mobius Driving Environment Build (Windows).....	13
2.2.1.	MySQL Installation.....	14
2.2.2.	MQTT Server Installation.....	22
2.2.3.	Node JS Installation.....	26
2.3.	Mobius Installation.....	29
2.3.1.	Download and Module Installation.....	29
2.3.2.	Mobiusdb Database Creation.....	31
3.	Mobius Server Platform Drive.....	37
3.1.	Environment Setting.....	37
3.2.	Test.....	39
3.2.1.	Resource Viewer(oneM2M Browser) Application.....	39
3.2.2.	Postman Application.....	41

1. Mobius Introduction

1.1. What is Mobius?

Mobius is a platform for managing various IoT device information providing IoT (Internet of Things) services based on oneM2M international standards, and providing access control, authentication, user management, and multiple IoT service combinations of these devices.

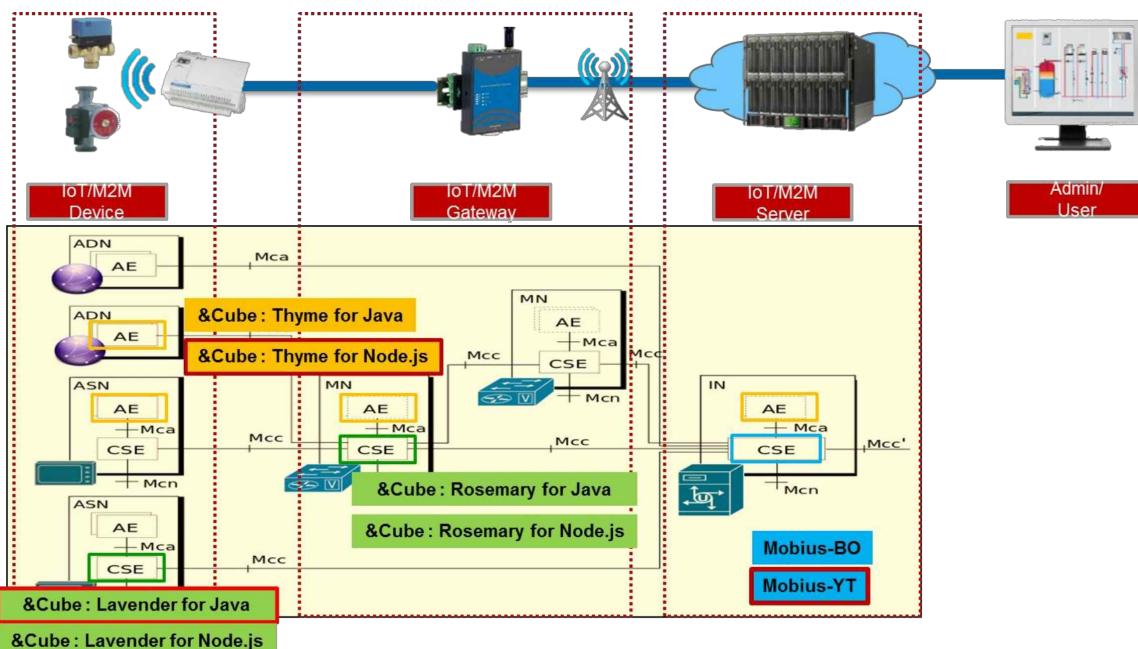


Figure 1 oneM2M-based Mobius Platform

Conceptual Diagram

The Mobius utilizes REST APIs (HTTP, MQTT, COAP, Websocket) to work with IoT devices as shown in the following figure2.

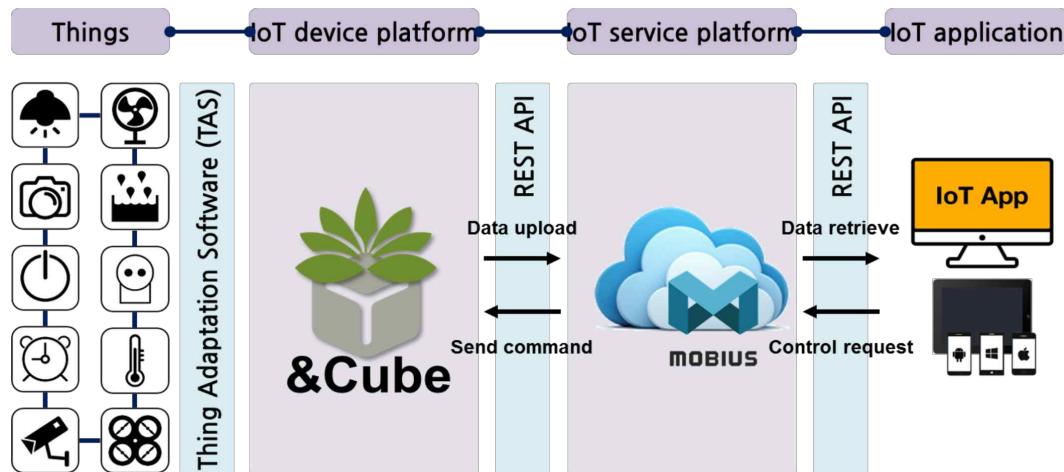


Figure 2 Mobius and IoT Device Interworking

When a resource is created, the Mobius automatically generates an API that can access that resource, through a REST interface, called the OPEN API.

1.2. Mobius Platform

1.2.1. Mobius Server Platform Outline

Mobius Platform is a server software platform that uses means of communication to manage things connected to the Internet and store data on these things, and can be said to be a kind of middleware. The Mobius Server Platform establishes connections between devices and applications with ease and convenience, and helps build an ecosystem where anyone can create and use IoT services in an open development environment.

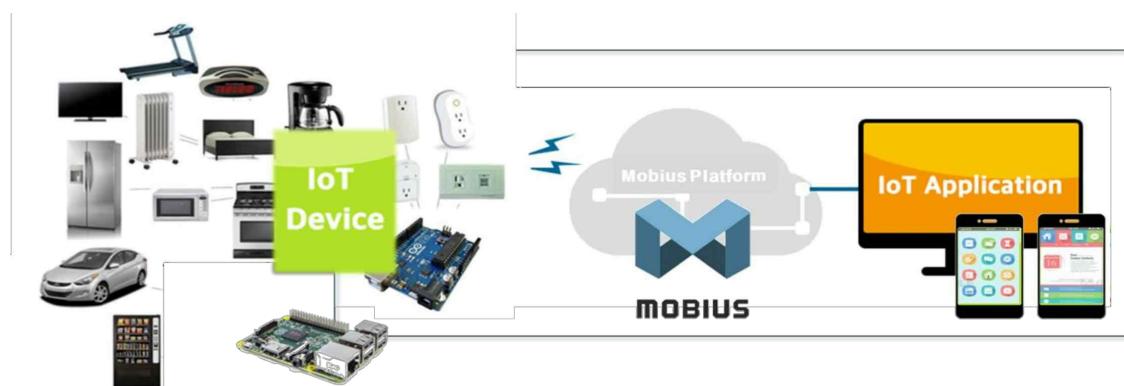


Figure 3 Mobius Platform Configuration

Mobius is a server platform developed in the JavaScript language using Node.js and is developed in accordance with the oneM2M international standard. Mobius development has been carried out using an express module.

Mobius supports HTTP, MQTT, COAP, WebSocket protocol and Open API in compliance with the oneM2M standard. According to the oneM2M standard, Mobius data is defined in a hierarchical resource structure, and can be accessed using a REST API method. The Mobius consists of an HTTP server for the HTTP Open API, an MQTT server for MQTT support, a COAP server for COAP, and a WebSocket server for WebSocket support, along with MySQL DBMS for resource storage.

1.2.2. Mobius Server Platform Key Features

The Mobius is a kind of middleware for connecting devices and applications, where the devices transmit data to the server platform, and the platform stores the data. The application queries the data stored on the platform through the Open API and transmits a control request to the platform. The platform transmits an application control request to the device again according to the set device.



Figure 4 Mobius Server Platform Key Features

1.2.3. Mobius Server Platform Configuration

Mobius connects devices and applications, supporting HTTP, COAP, MQTT interfaces, WebSocket protocols for connection with devices, and the DataBase(DB) for resource storage, configured using MySQL.

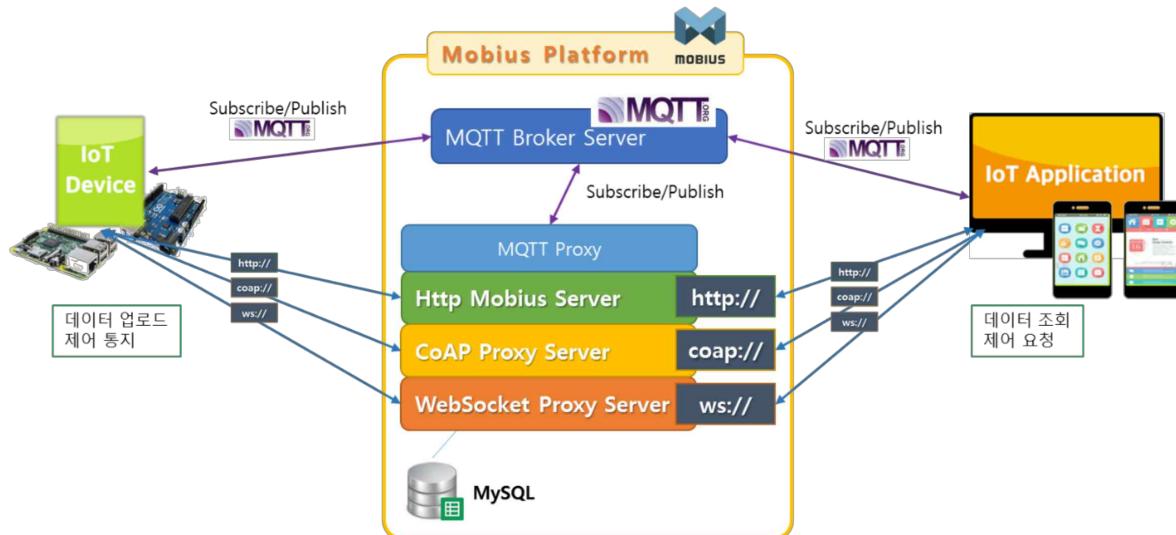


Figure 5 Mobius Server Platform System Configuration

1.2.4. Mobius Server Platform S/W Architecture

Mobius was developed as a JavaScript language based on Node.JS. The DB supports MySQL and communication protocols support HTTP, MQTT, COAP, and WebSocket. The following figure6 shows Mobius' software architecture.

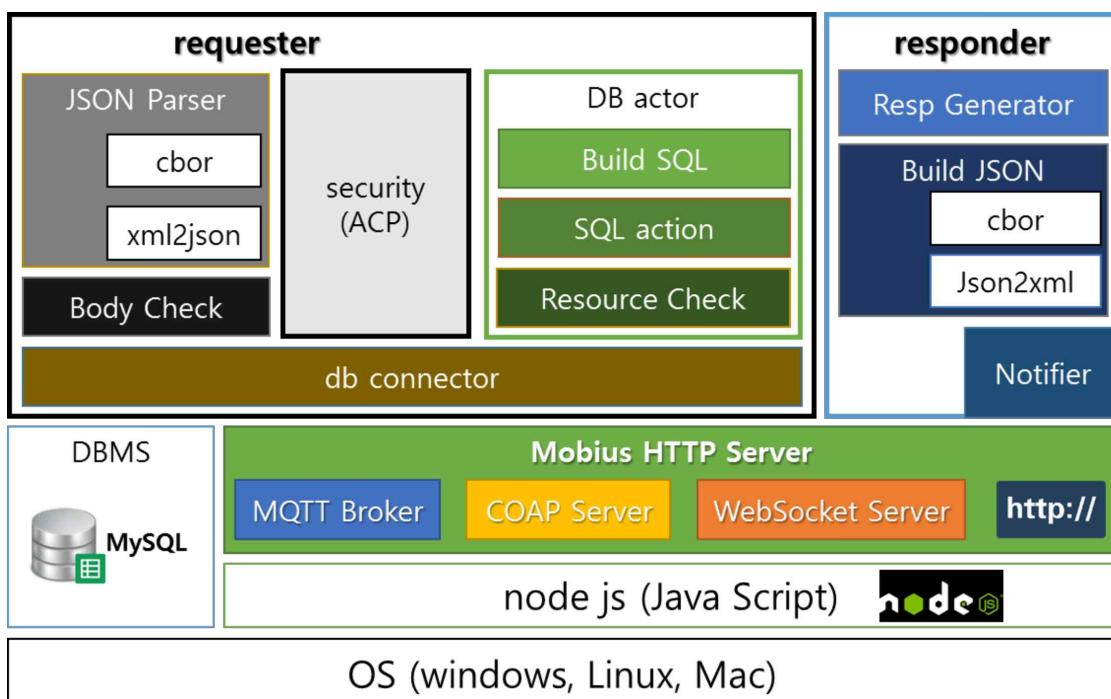


Figure 6 Mobius S/W Architecture

Mobius basically builds MQTT brokers, COAP servers, and WebSocket servers and http servers, and provides Mobius HTTP Server as an overlay for these servers.

As shown in the figure above, to support MQTT, a proxy for converting to MQTT and HTTP is implemented internally, and likewise, communication for COAP and WebSocket are supported as well.

Mobius consists of two main parts: requester and responder, where the requester contains components for DB access. All HTTP requests are delivered to the requestor component, sql statements are made to access data through parser-actor, and data is generated, viewed, and searched through the db connector. When the DB data access result is released, the data format: xml, json or cbor is determined in response to the request and processed by the responder.

Mobius supports a tree-like resource scheme such as Windows File Explorer. The resource scheme maps each resource directly to a URI that can access the resource, similar to accessing a folder in the File Explorer. For example, the URL for accessing the data under IN-CSE, under AE, and the Container is as follows.

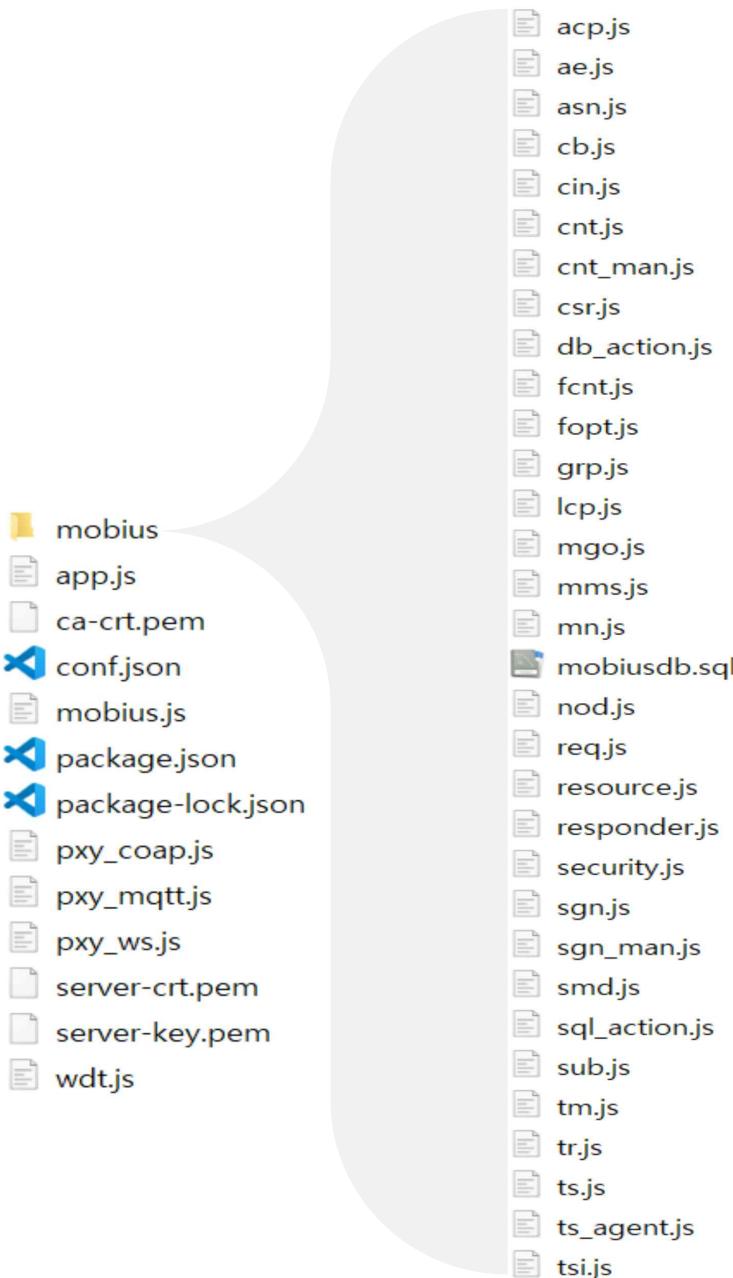
http://hostname:port/IN-CSE_name/AE_name/container_name/contentInstance_name

As such, all resources can be accessed, and CREATE, READ, UPDATE and DELETE (CRUD) operations can be performed through POST, GET, PUT, and DELETE http methods. (see OCEAN's API documentation)

A new container resource can be defined within a Container, where the depth of the resource URL is not limited.

1.2.5. Mobius Server Directory Configuration

The following figure shows the directory structure of the Mobius platform.



The Mobius project structure mostly consists of the “mobius” folder and the mobius.js, app.js, pxy_mqtt.js, pxy_coap.js, pxy_ws.js, wdt.js, conf.json, package.json, and openslkey files. The “mobius” folder contains several source files, including db_action.js, ft.js, resource.js, responder.js, security.js, sgn.js, sql_action.js, ts_agent.js, as well as the source files that play specific roles. The specific functions and roles for each file are shown in the table below.

Source File	Functions and Roles
mobius.js	The attribute value of this script/file can be configured for the proper execution of Mobius. The main code of Mobius can be loaded and executed through this file.
app.js	A file that acts as a main script, and all packets entering Mobius are processed first in this file. Run the http server on this file, listen, and wait. When a packet is received, it parses the target url of the packet, analyzes the body of the packet, determines whether it is a normal packet, and when it is determined as normal packet, it forwards it to the resource.js that performs the corresponding operation. In case of an error packet, it returns the error through responder.js. In this file, the http server is implemented to run the server through clustering to improve server performance.
pxy_mqtt.js	This script processes mqtt-related data as a file that binds mqtt to http. In the beginning, it takes the mobius information to create and subscribe to oneM2M mqtt topic. When the data is received by mqtt, it is converted into an http packet and forwarded to Mobius, and after receiving the response, it is converted back into an mqtt packet and mqtt response is returned.
pxy_coap.js	This script processes coap data as a file that binds to http. At startup, a coap server is created, and subsequently, when coap data is received, a corresponding http packet is formed and forwarded to Mobius, and upon receiving a response, it is converted back into a coap packet and is sent back as a response by coap.
pxy_ws.js	This script processes websocket data as a websocket to http binding file. At startup, a websocket server is created, and when websocket data is received, it is converted into an http packet and forwarded to Mobius, and when a response is received from Mobius, it is converted back into a websocket packet and responded to the websocket.
wdt.js	This script is built with a watchdog timer of Mobius and is used as a timer for Mobius.

mobius/acp.js	accessControlPolicy resource file – resources related to access control
mobius/ae.js	AE resource file
mobius/asn.js	Application Service Node resource file
mobius/cb.js	CSEBase resource file - Mobius has its own information.
mobius/cin.js	contentInstance resource file – The actual data is stored here.
mobius/cnt.js	container resource file
mobius/cnt_man.js	Files that assist the role of cnt.js
mobius/csr.js	remoteCSE resource file
mobius/db_action.js	A file containing a function that connects to a real database and accesses the database and returns results.
mobius/fcnt.js	flexContainer resource file
mobius/fopt.js	Responsible for processing fanOutPoint resources for group resources. The fanOutPoint resource, as a virtual resource, delivers the requested packet to all the members included in the group resource, collects the results, and responds to the requestor.
mobius/grp.js	group resource file
mobius/lcp.js	locationPolicy resource file
mobius/mgo.js	mgmtObj resource file
mobius/mms.js	multimediaSession resource file
mobius/mn.js	Middle Node resource file
mobius/mobiusdb.sql	List of each resource defined in the database
mobius/nod.js	node resource file
mobius/req.js	request resource file – nonblocking Feature Support
mobius/resource.js	This file is a core file that performs create, read, update, delete, notify, and discovery on all resources supported by Mobius. The url body verified in app.js is converted into an internally processed format according to the corresponding method and the actual database interworking is performed. The database behavior is based on db_action.js and It is performed via sql_action.js.
mobius/responder.js	This file is responsible for returning back the results processed by app.js, resource.js for the corresponding request packet. When responding, the response data is

	modified according to the json and xml format and then returned.
mobius/security.js	This file checks the accessControlPolicy for the target resource to determine whether to allow or reject the request. The accessControlPolicy resource is checked, which is connected to the target resource, having the id value contained in the X-M2M-Origin header of the request packet to verify that the id has access.
mobius/sgn.js	This script is responsible for checking whether a subscription exists as a child on a target resource, and, if a subscription exists, it checks the event, and then creates and sends a request message based on the notificationUri attribute information in the subscription.
mobius/sgn_man.js	Script that assist the role of sgn.js
mobius/smd.js	semanticDescriptor resource file
mobius/sql_action.js	This script creates an sql statement to obtain the data needed for Mobius operations in the database, and then returns and transfers the data from the database through a function of db_action.js.
mobius/sub.js	subscription resource file
mobius/tm.js	transaction management file
mobius/tr.js	transaction resource file
mobius/ts.js	timeSeries resource file
mobius/ts_agent.js	This file is responsible for managing timeSeries Instance resources, which periodically monitors the data that is missing in the timeSeries Instance and restores the data.
mobius/tsi.js	timeSeriesInstance resource file

2. Mobius Server Platform Installation

2.1. Installation Outline

Mobius Platform uses MySQL DBMS. If you install MySQL first and install MQTT Broker and NodeJS in the order of installation, the environment for running the Mobius server is completed, and then the Mobius server is installed to complete the platform construction.



Figure 7 Mobius Server Platform Installation
Outline

2.2. Mobius Driving Environment Build (Windows)

2.2.1. MySQL Installation

Mobius operates based on MySQL DB for data storage and management. This section describes how to download and install MySQL, an open-source RDBMS.

General Availability (GA) Releases Archives 

MySQL Community Server 8.0.29

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:

File Type	Version	Size	Action
Windows (x86, 64-bit), ZIP Archive (mysql-8.0.29-winx64.zip)	8.0.29	214.0M	
Windows (x86, 64-bit), ZIP Archive Debug Binaries & Test Suite (mysql-8.0.29-winx64-debug-test.zip)	8.0.29	538.6M	

 We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

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

If you go to MySQL, you can download the free version of MySQL Community Server by clicking on the link above. To install MySQL on Windows, click the Download button that says Windows (x86, 64-bit), MySQL Installer MSI. (Please note that MySQL has some commercial features such as MySQL Enterprise Edition and MySQL Cluster CGE.)

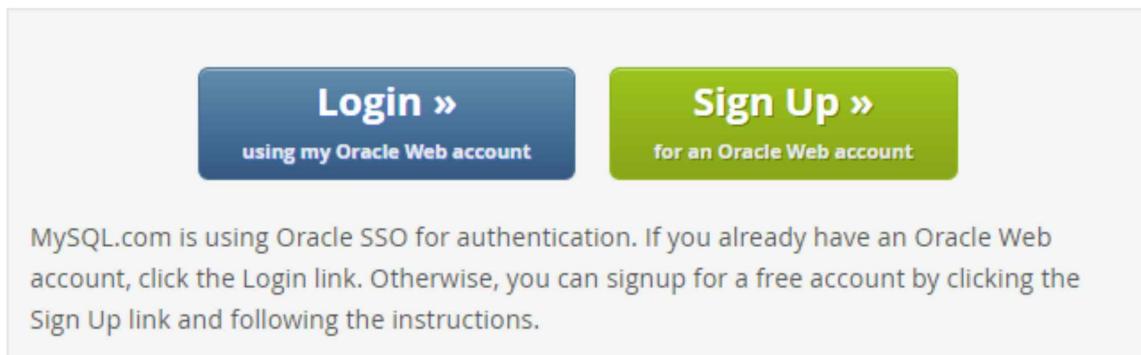
Click the Zip Archive download button to download and install all of the installation files(not the web installation). This version can be used to install MySQL in places where web installation is difficult.

④ MySQL Community Downloads

[Login Now](#) or [Sign Up](#) for a free account.

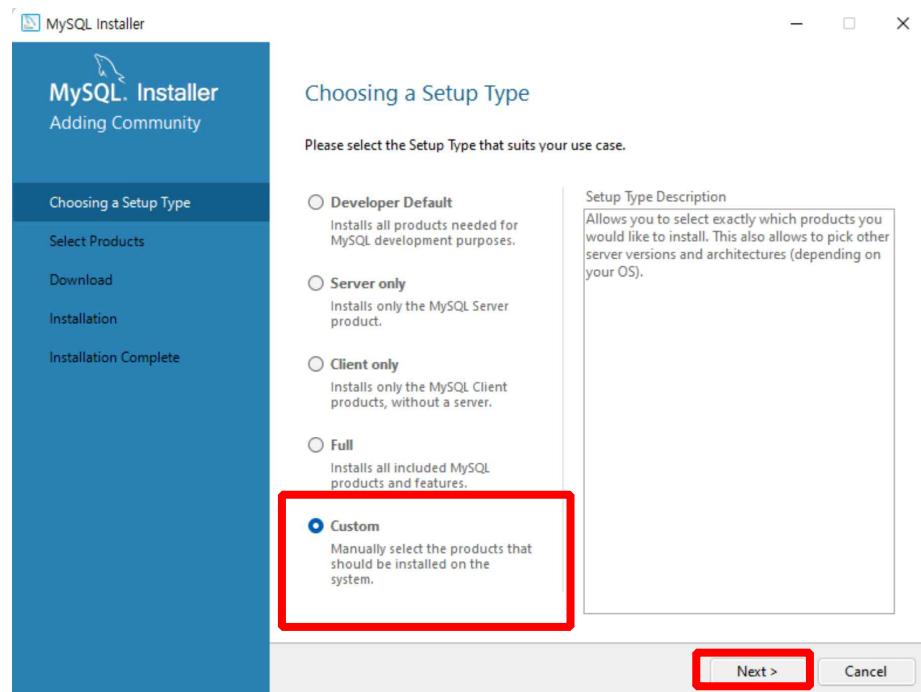
An Oracle Web Account provides you with the following advantages:

- 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

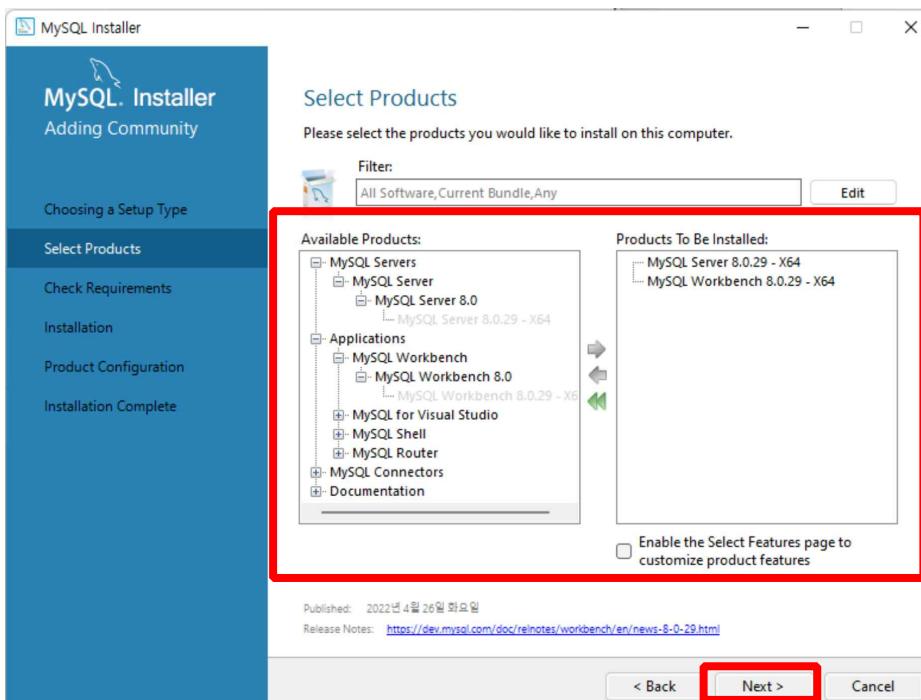


When the login screen appears, sign up for membership or click the **No thanks, just start my download** link below to download it.

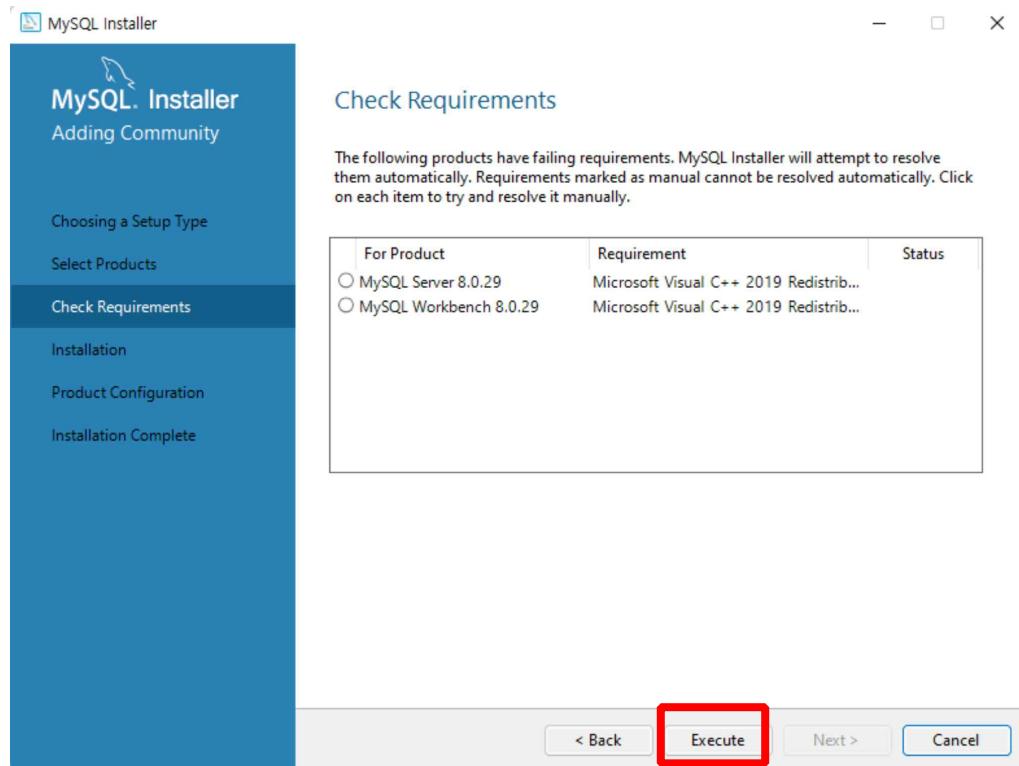
Download and run the MySQL installation file to agree to the license and press the Next button.



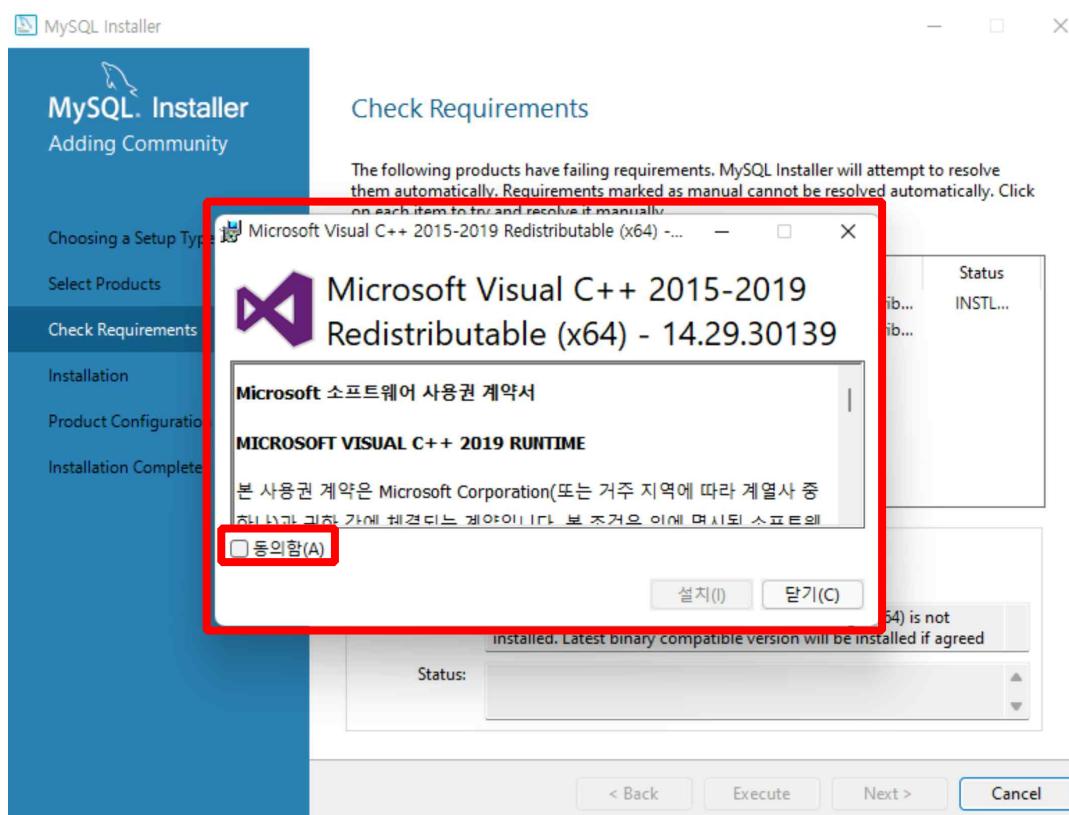
There are five installation styles: "Developer Default", "Server only", "Client only", "Full", and "Custom". The Mobius requires MySQL Server and MySQL Workbench. Select Custom here and press the Next button to proceed with the installation.



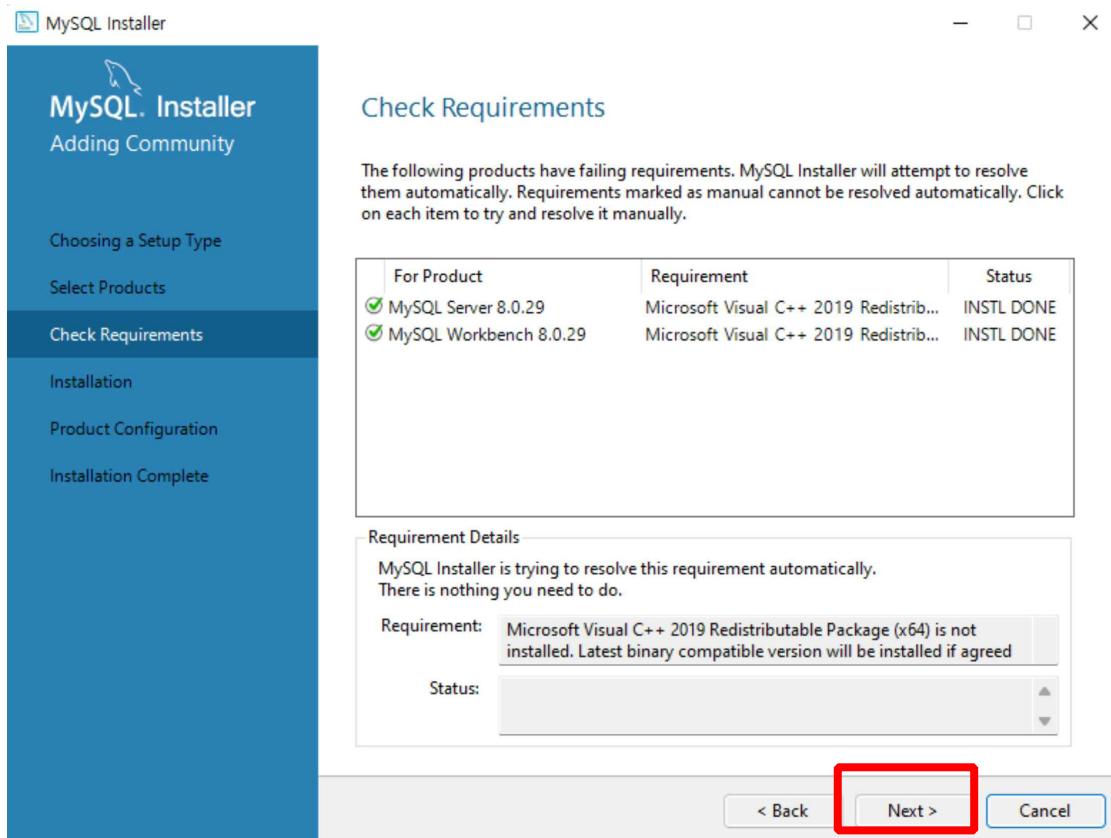
Select MySQL Server and MySQL Workbench. Click Next.



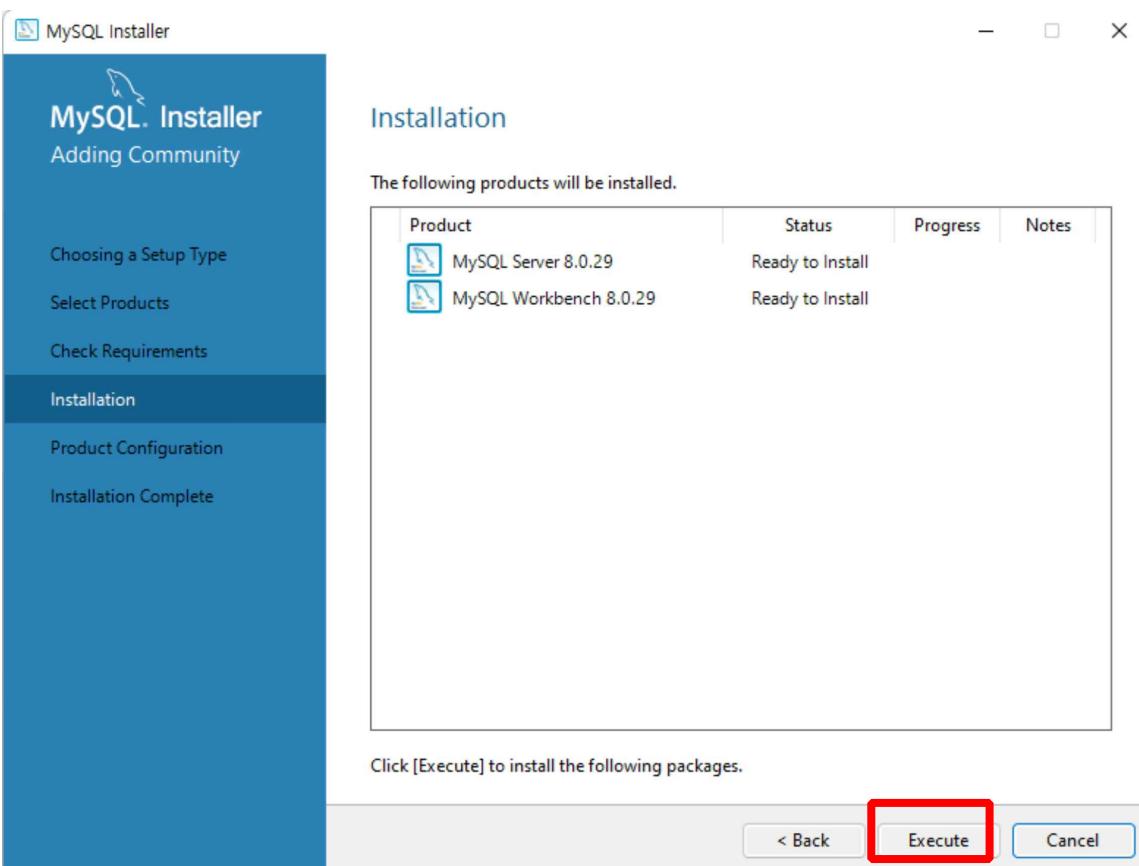
Click "Execute" to install MySQL Server and MySQL Workbench.



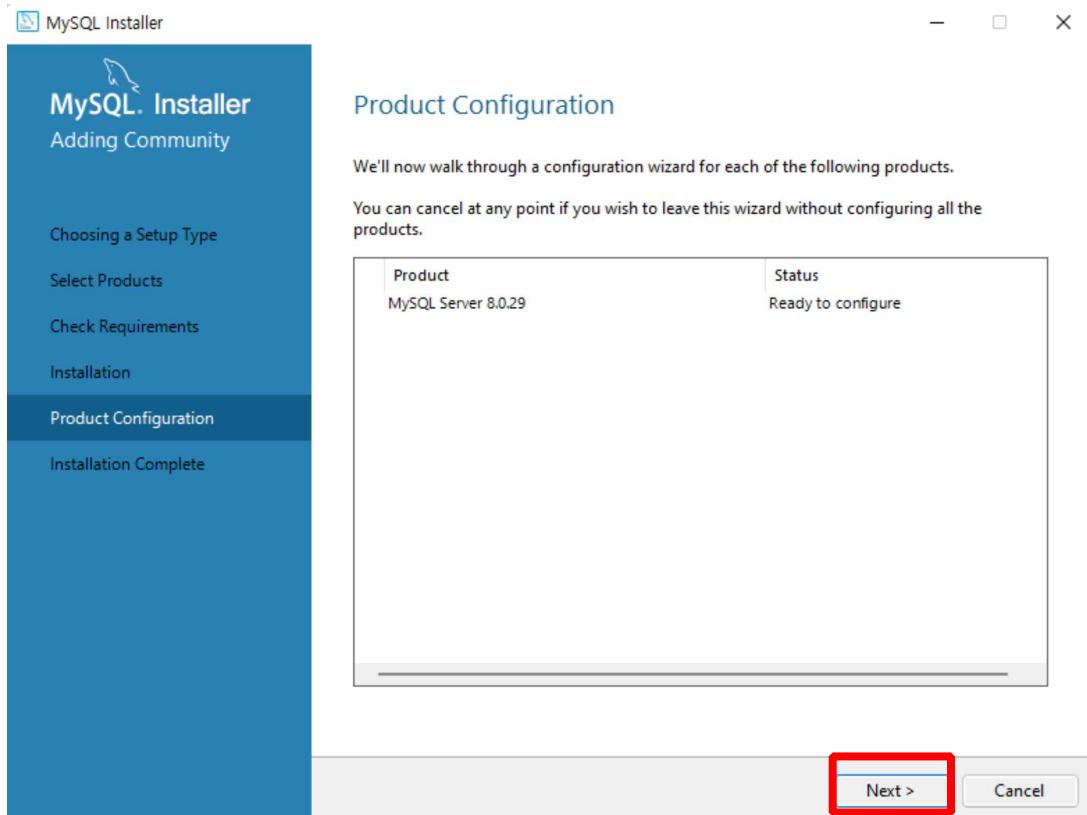
If Visual C++ is not installed, a window for installing the C++ runtime component appears as above, and you can proceed with the installation after agreeing to the license. If Visual C++ is already installed, the above window does not appear, so proceed to next step.



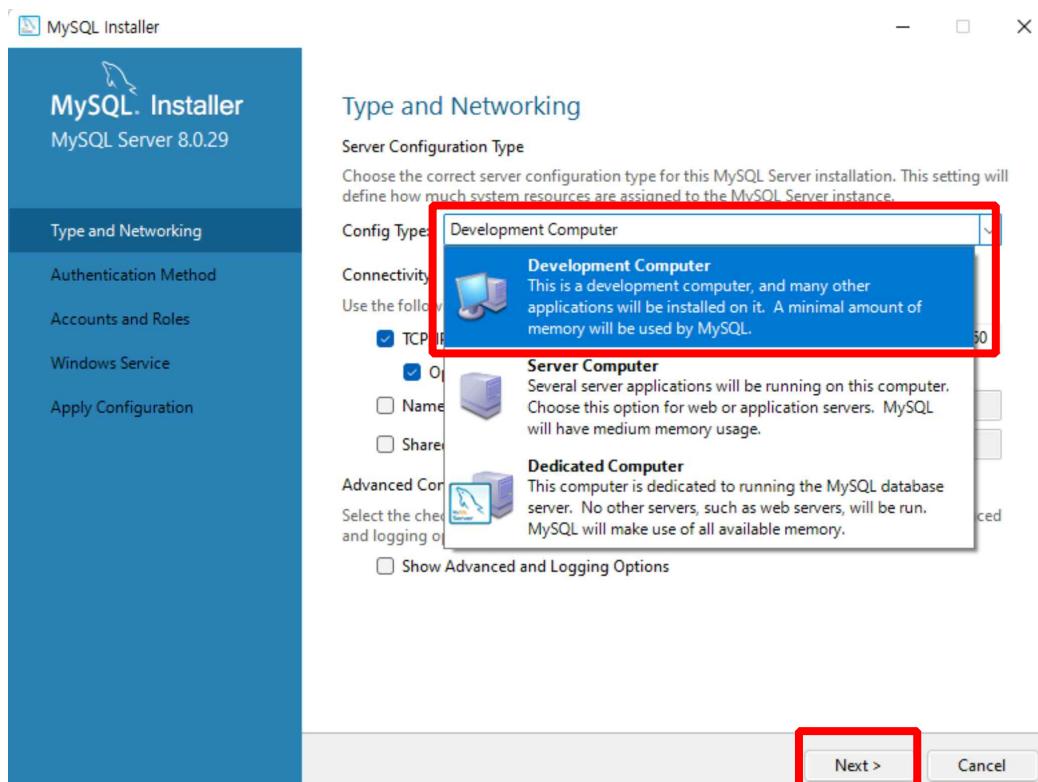
Click "Next" for installation.



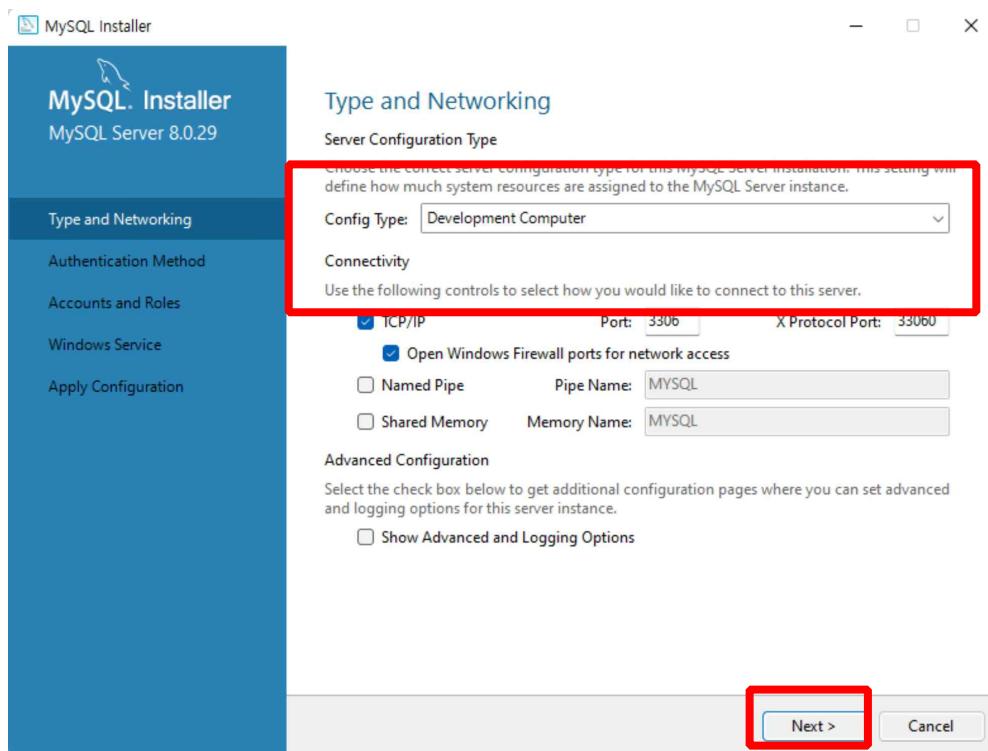
Click "Execute" for installation.



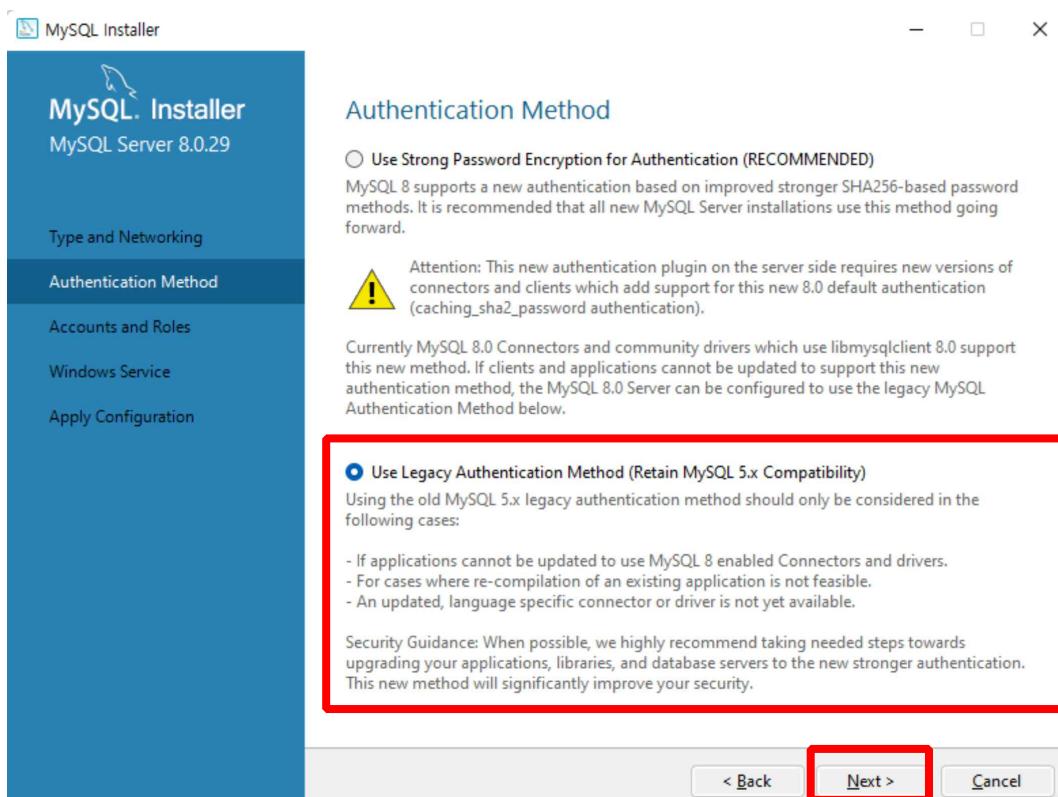
When MySQL installation is complete, click "Next" for configuration tasks.



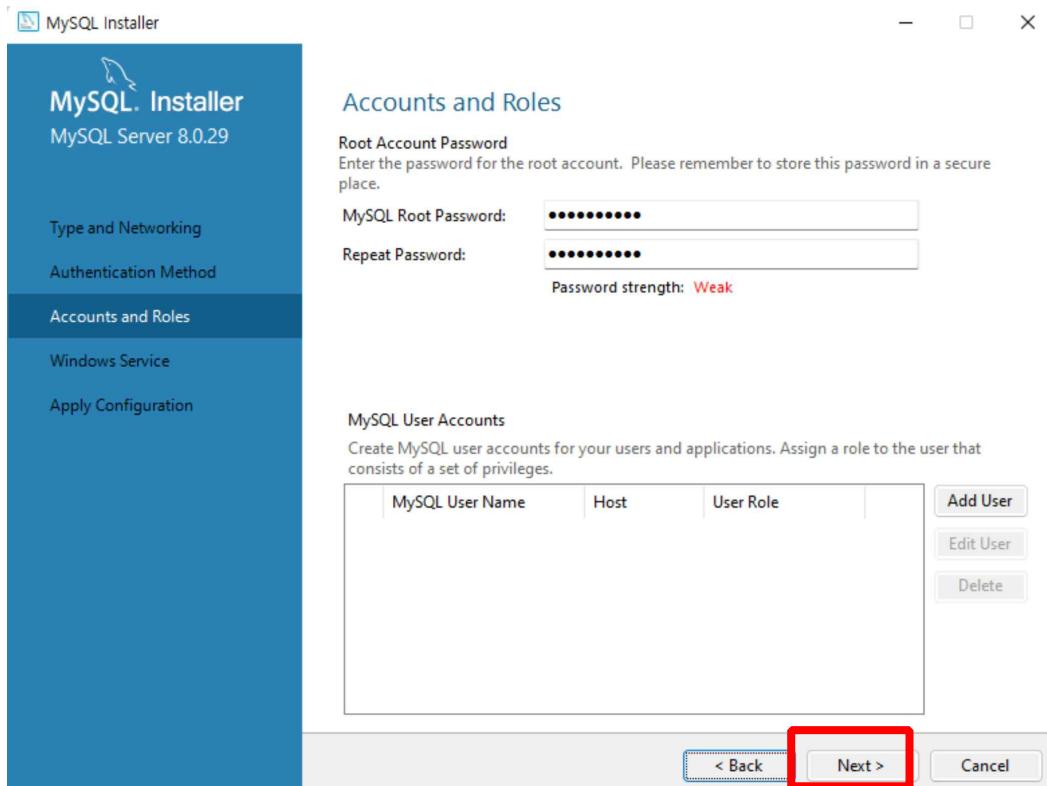
The configuration types include "Development Machine", "Server Machine", and "Dedicated Machine", where you can choose to be a developer and select "Development Machine". Click "Next".



The default port for MySQL is 3306. Click "Next".

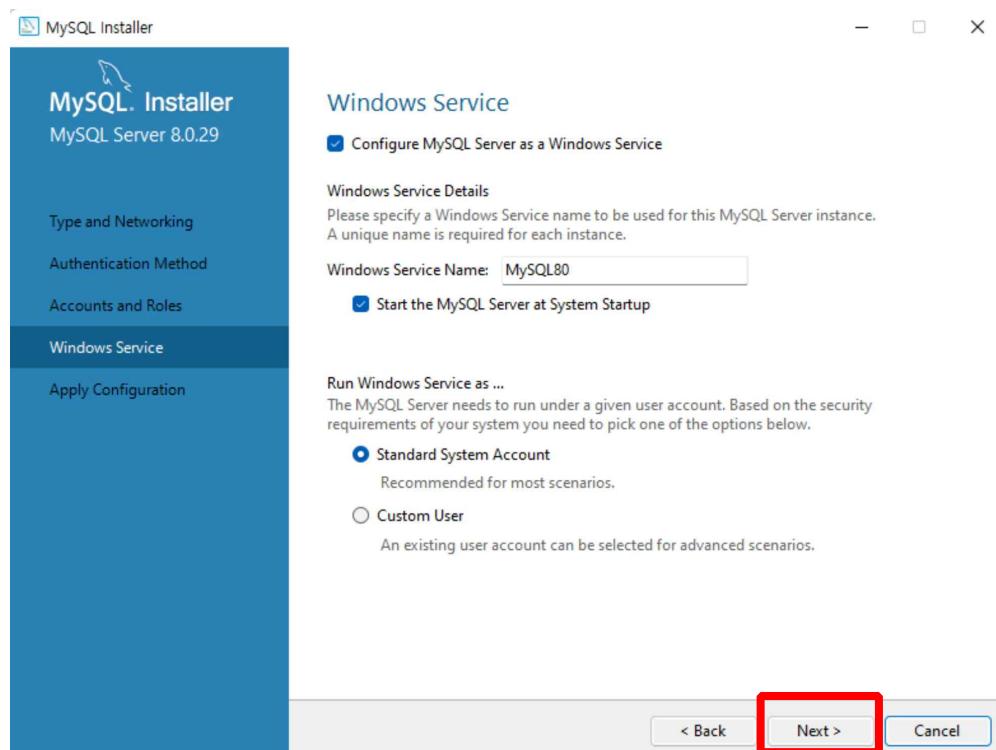


In order to be compatible with the MySQL 5.x version and the authentication process, installation must be performed with the Legacy Authentication Method.

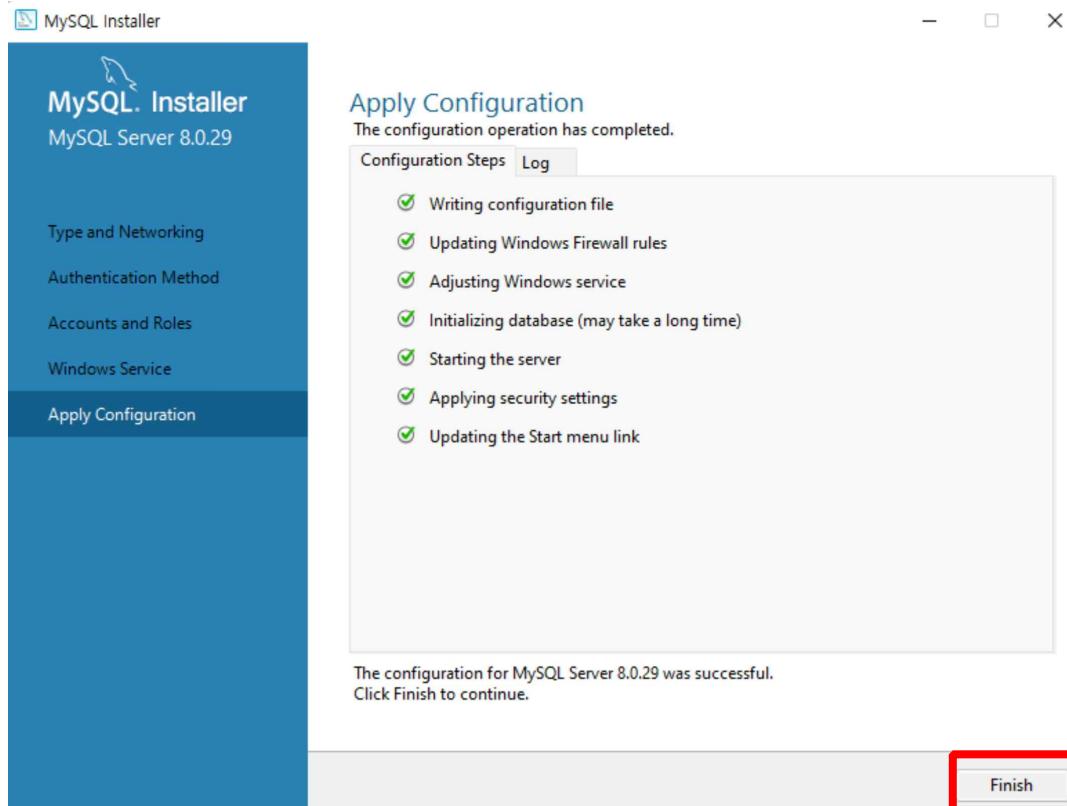


Specify the "MySQL root password" and click "Next".

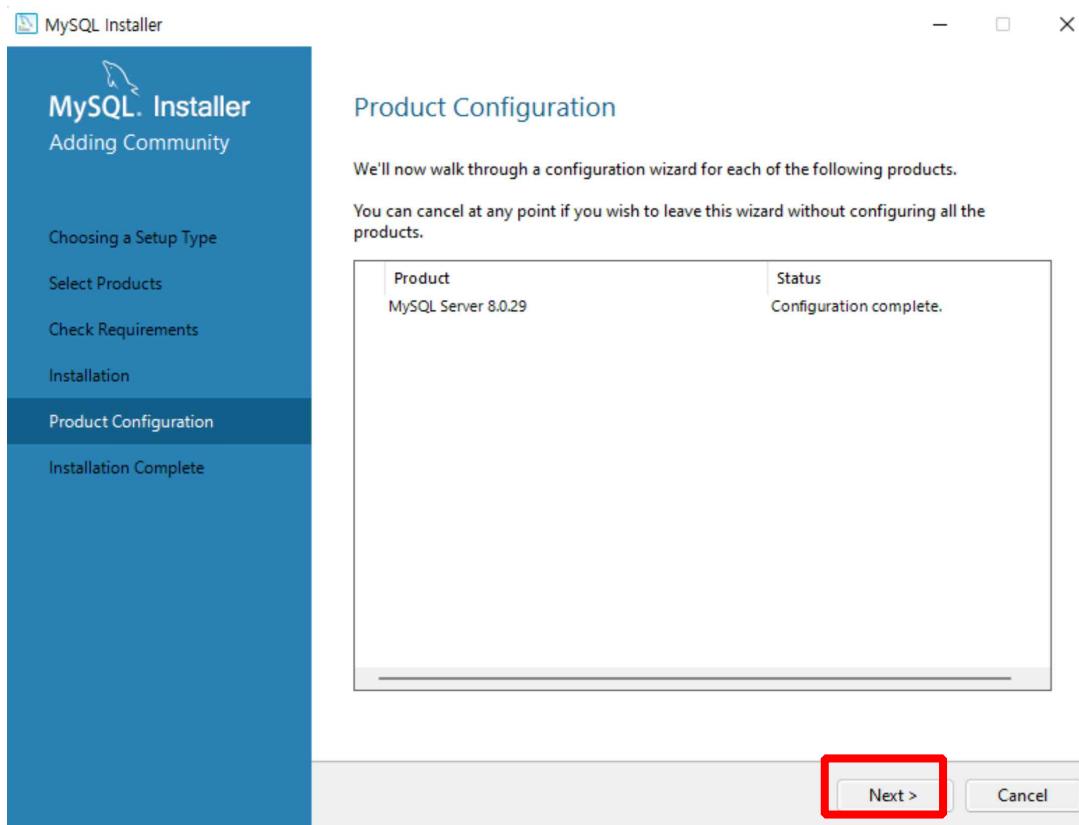
It is required when linking Mobius and Mysql.



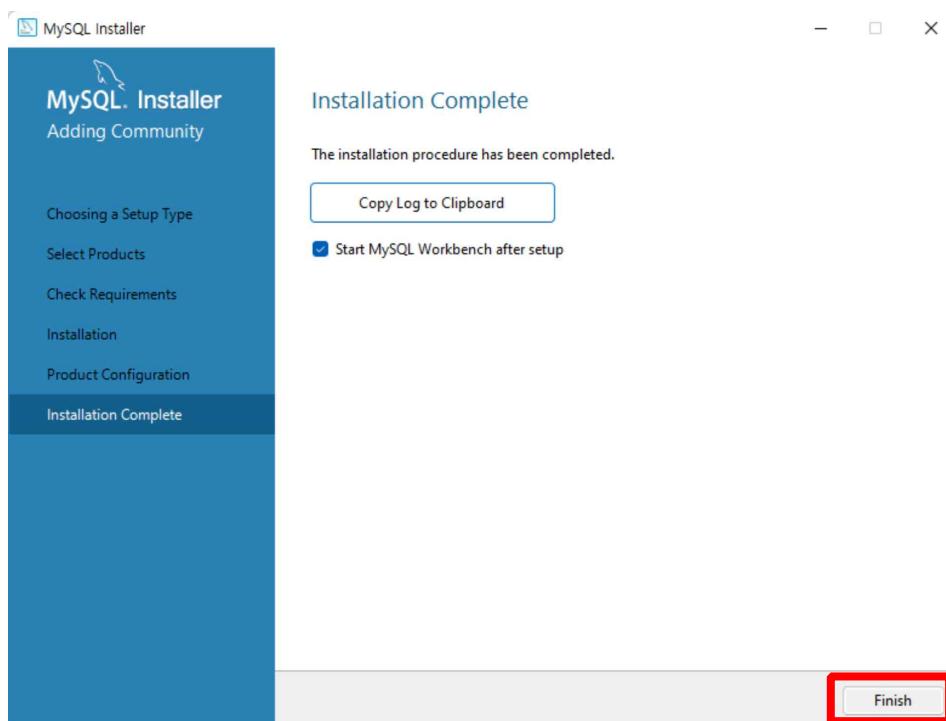
Specify the Windows service name for the MySQL Service instance to be executed when Windows boots. Click "Next".



Verify that the MySQL Server configuration is set correctly and press "Finish".



Click "Next" for other preferences.



MySQL installation is complete. Press the "Finish" button to complete the installation.

2.2.2. MQTT Broker Installation

MQTT(Message Queuing Telemetry Transport)is an publish/subscription-based messaging protocol.

Mobius supports the MQTT protocol for communication with devices, and to this end, it installs the Mosquitto server, an open source MQTT broker.

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

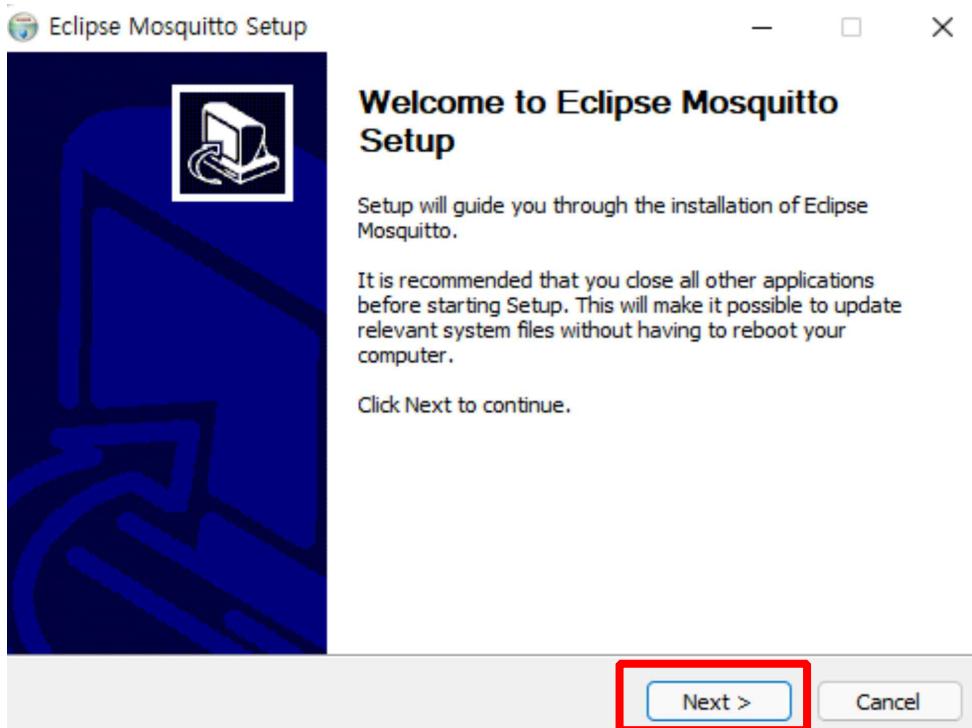
Download and install the mosquito installation file by pressing the download menu on the site and selecting the Windows installation file using the Binary Installation (mosquitto-1.4.3-install-win32.exe).

The current version has additional work to do, and the installation method can be easily found by searching the "Mosquitto Installation" on the internet and can be copied and installed.

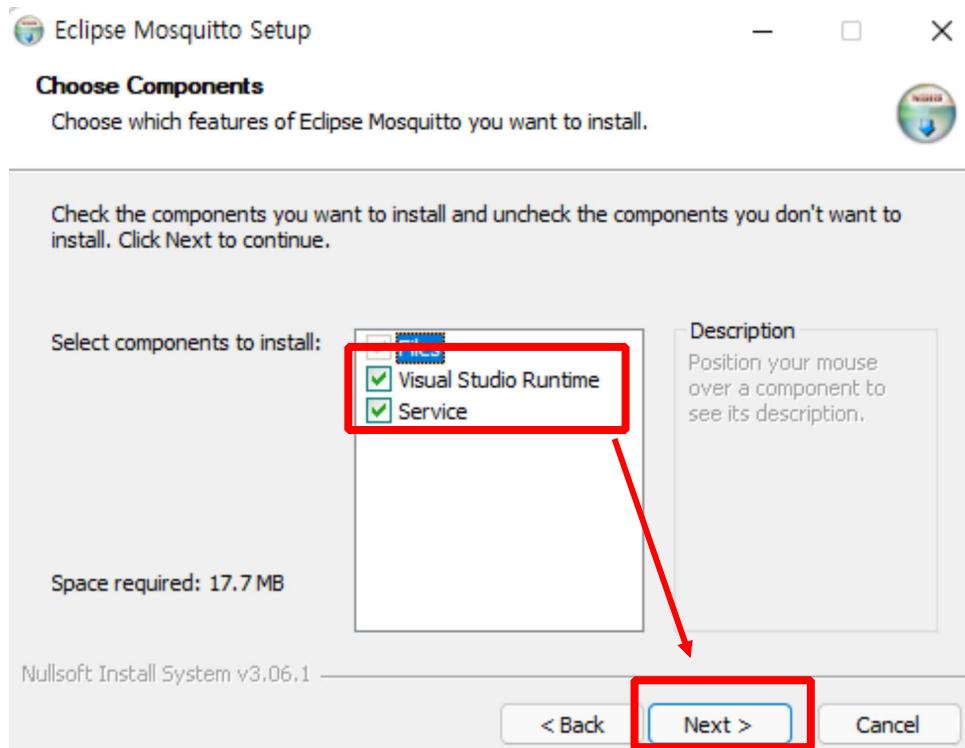
This document describes how to install an earlier version without additional work to do.

The screenshot shows the Mosquitto download page. At the top, there are links for mosquito, ECLIPSE FOUNDATION, and cedalo. Below these are links for Home, Blog, Download, and Documentation. A 'Download' button is prominently displayed. Under the 'Source' heading, there are links for mosquito-2.0.14.tar.gz (GPG signature) and Git source code repository (github.com). A note says 'Older downloads are available at <https://mosquitto.org/files/>'. The 'Binary Installation' section is titled 'Windows' and contains two download links: mosquito-2.0.14-install-windows-x64.exe (64-bit build, Windows Vista and up, built with Visual Studio Community 2019) and mosquito-2.0.14-install-windows-x32.exe (32-bit build, Windows Vista and up, built with Visual Studio Community 2019). The 'Windows' link is highlighted with a red box.

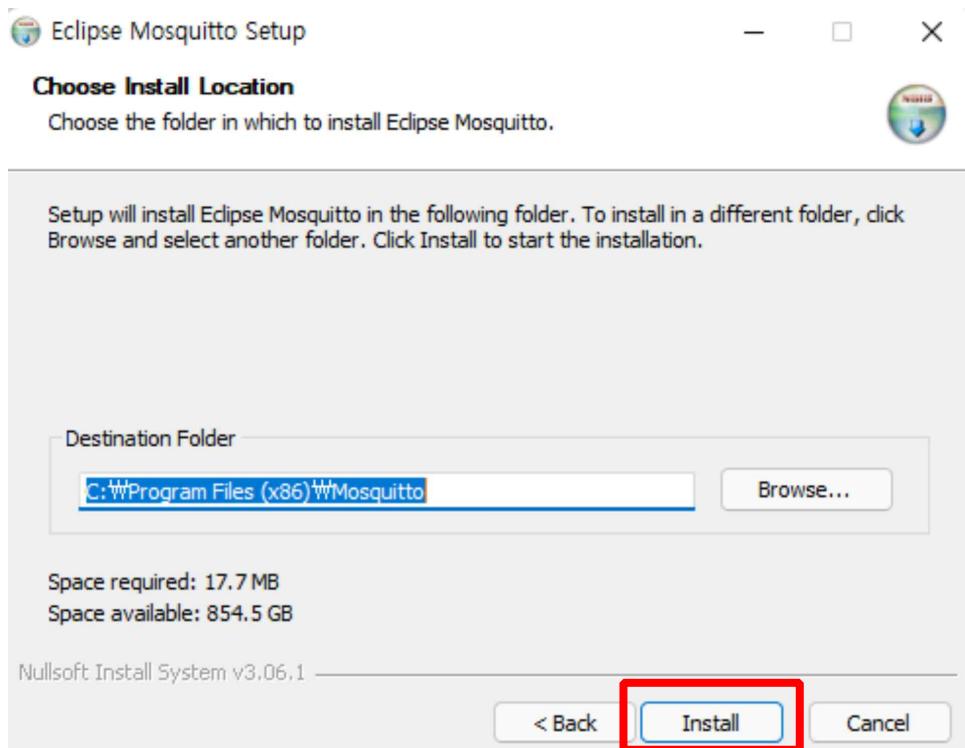
Download the installation file according to your operating system.



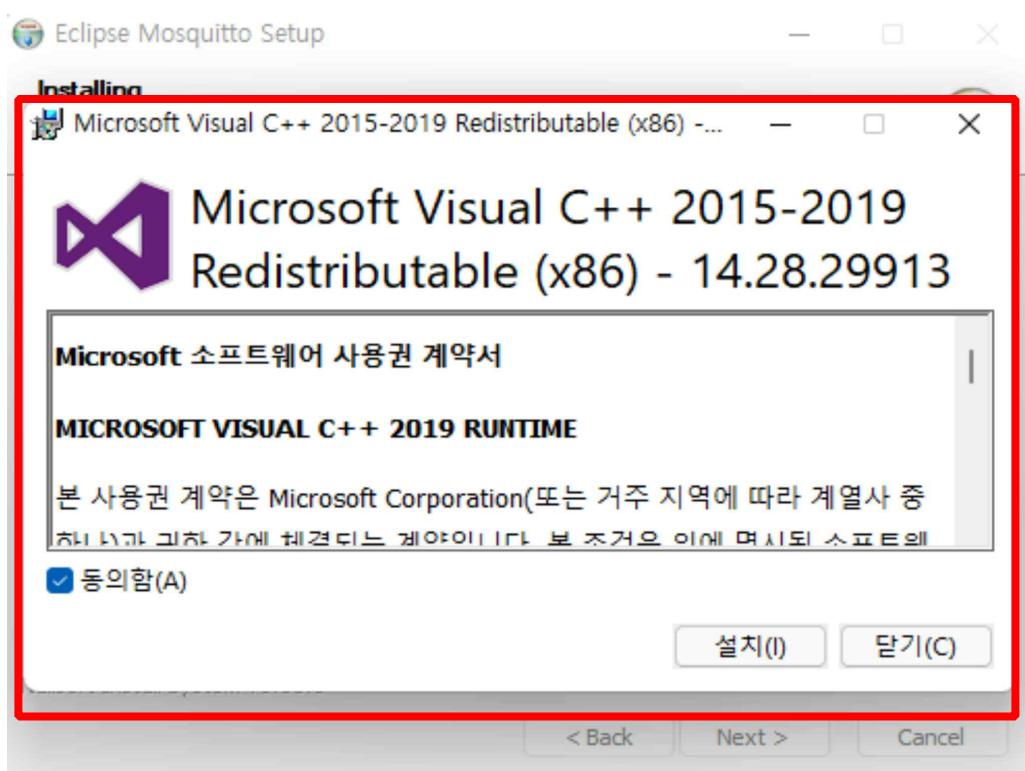
Click Next for installation.



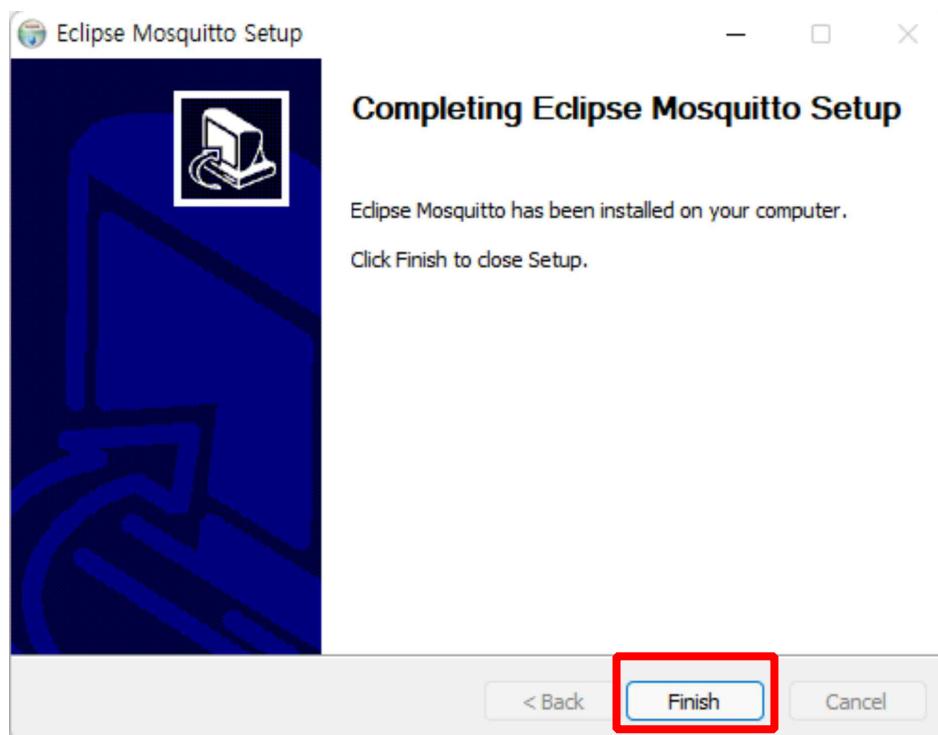
During installation, the option to start the MOSQUITO server as a Windows service appears as shown in the figure above, and leave it as default and press "Next". After installation, it is registered as a Windows service without any additional work, and the MOSQUITO server starts automatically.



Click "Install" for installation.



If Visual C++ is not installed, a window for installing the C++ runtime component appears as above, and you can proceed with the installation after agreeing to the license. If Visual C++ is already installed, the above window does not appear, so you can proceed to next step.



Click the "Finish" button when the installation is complete. It will be installed normally, but you can check the normal operation of the MQTT broker MOSQUITO server as follows.

Once the installation is complete, you must start Mosquitto, which is registered as a Windows service. You can find Mosquitto in the Services tab by opening Windows Task Manager as shown in the following figure.

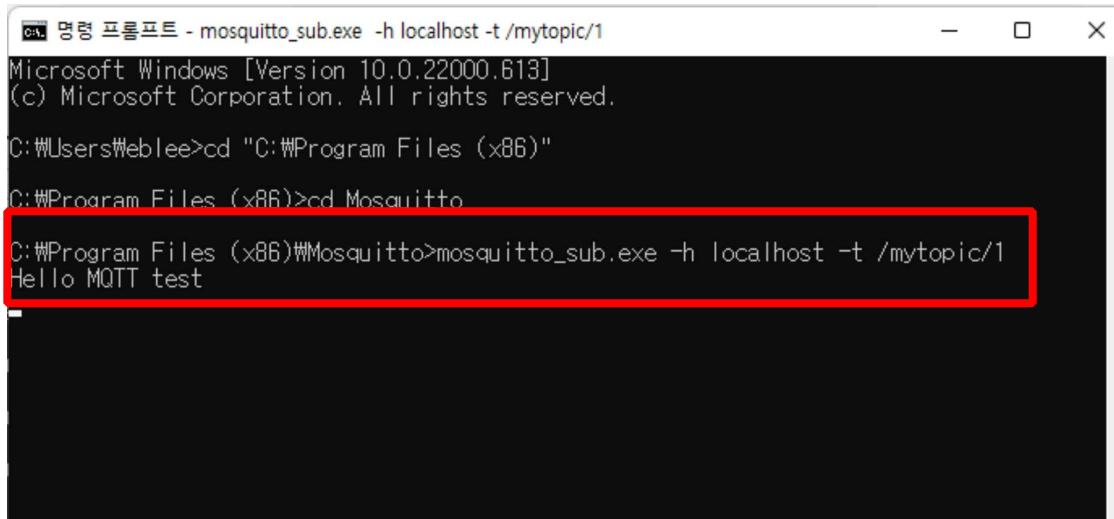
작업 관리자			
프로세스	성능	앱 기록	시작프로그램
이름	PID	설명	상태
MySQL57	4144	MySQL57	실행 중
MyFw40Service		MyFirewall 4.0 Service	중지됨
msiserver		Windows Installer	중지됨
MSDTC		Distributed Transaction...	중지됨
MozillaMaintenance		Mozilla Maintenance S...	중지됨
mosquitto		Mosquitto Broker	중지됨
MaWebDRMSVC	4156	Markany WebDRM No...	실행 중
LMS	1868	Intel(R) Management a...	실행 중
KOS_Service	4124	Kings Online Security	실행 중
KeyIso	780	CNG Key Isolation	실행 중
jhi_service	564	Intel(R) Dynamic Appli...	실행 중
IpOverUsbSvc	3992	Windows Phone IP ove...	실행 중
Intel(R) Capability License		Intel(R) Capability Lic...	실행 중

작업 관리자			
프로세스	성능	앱 기록	시작프로그램
이름	PID	설명	상태
MySQL57	4144	MySQL57	실행 중
MyFw40Service		MyFirewall 4.0 Service	중지됨
msiserver		Windows Installer	중지됨
MSDTC		Distributed Transaction...	중지됨
MozillaMaintenance		Mozilla Maintenance S...	중지됨
mosquitto	26680	Mosquitto Broker	실행 중
MaWebDRMSVC	4156	Markany WebDRM No...	실행 중
LMS	1868	Intel(R) Management a...	실행 중
KOS_Service	4124	Kings Online Security	실행 중
KeyIso	780	CNG Key Isolation	실행 중
jhi_service	564	Intel(R) Dynamic Appli...	실행 중
IpOverUsbSvc	3992	Windows Phone IP ove...	실행 중
Intel(R) Capability License		Intel(R) Capability Lic...	실행 중

After the initial installation, Mosquitto will be stopped and you can start the service by right-clicking and pressing "Start".

To test whether Mosquitto is properly installed, run a command prompt and navigate to the folder where Mosquito server is installed as shown below to run the following command.

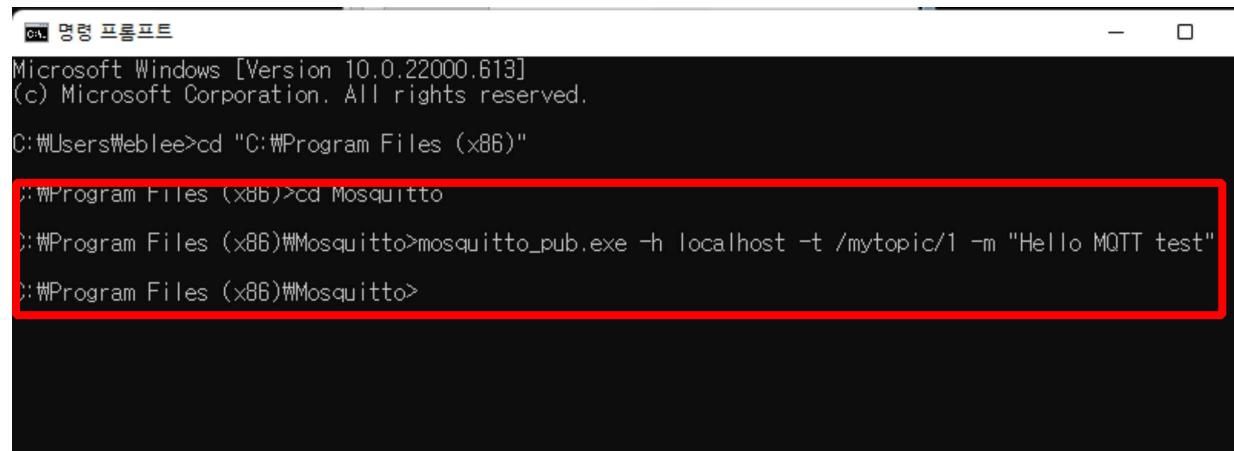
> *mosquitto_sub.exe -h localhost -t /mytopic/1*



```
명령 프롬프트 - mosquitto_sub.exe -h localhost -t /mytopic/1
Microsoft Windows [Version 10.0.22000.613]
(c) Microsoft Corporation. All rights reserved.

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
```

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



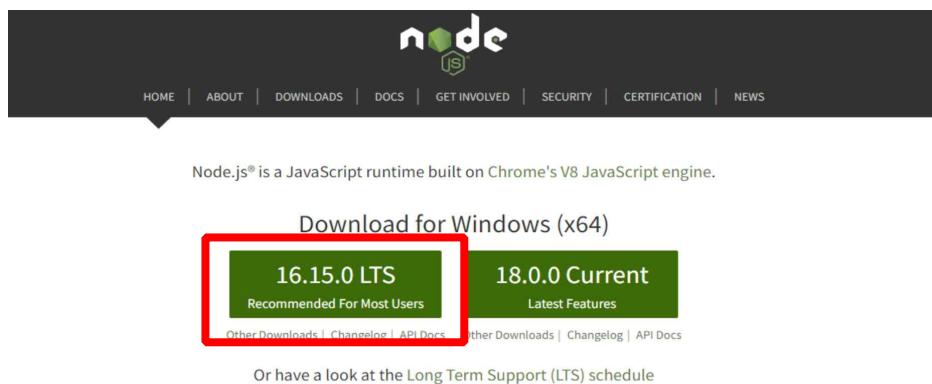
```
명령 프롬프트
Microsoft Windows [Version 10.0.22000.613]
(c) Microsoft Corporation. All rights reserved.

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>
```

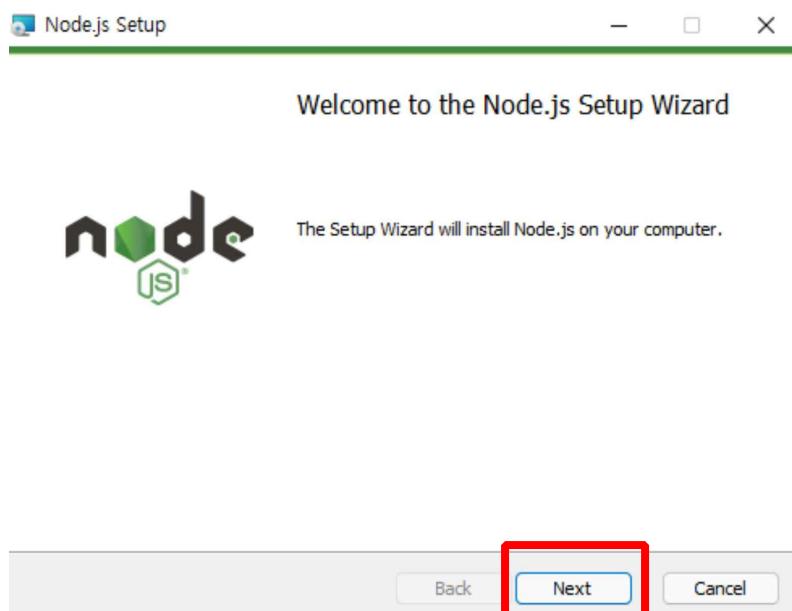
2.2.3. Node JS Installation

Node.js is a single-thread-based network server that supports high-performance asynchronous IO (Async/Non-blocking IO). Development began in 2009 by Ryan Dahl and it is now one of the open source projects with numerous support modules. Node.js is an event-based programming model developed by the Google Chrome V8 engine and uses JavaScript as the programming language. In recent years, many companies are adopting Node.js.

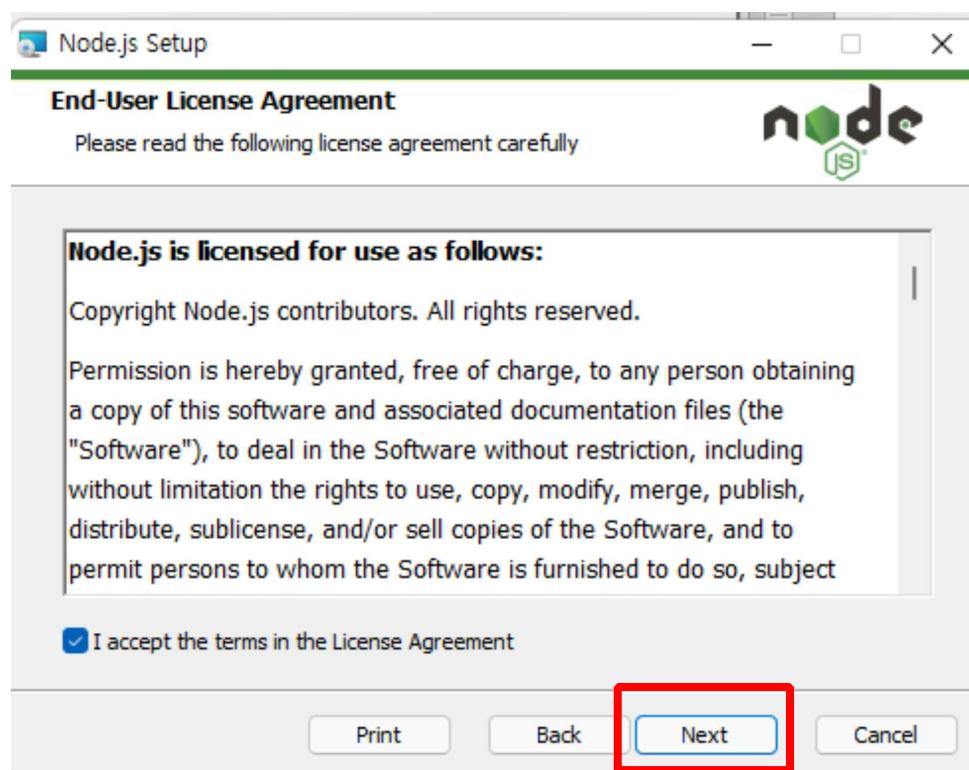
<http://www.nodejs.org>



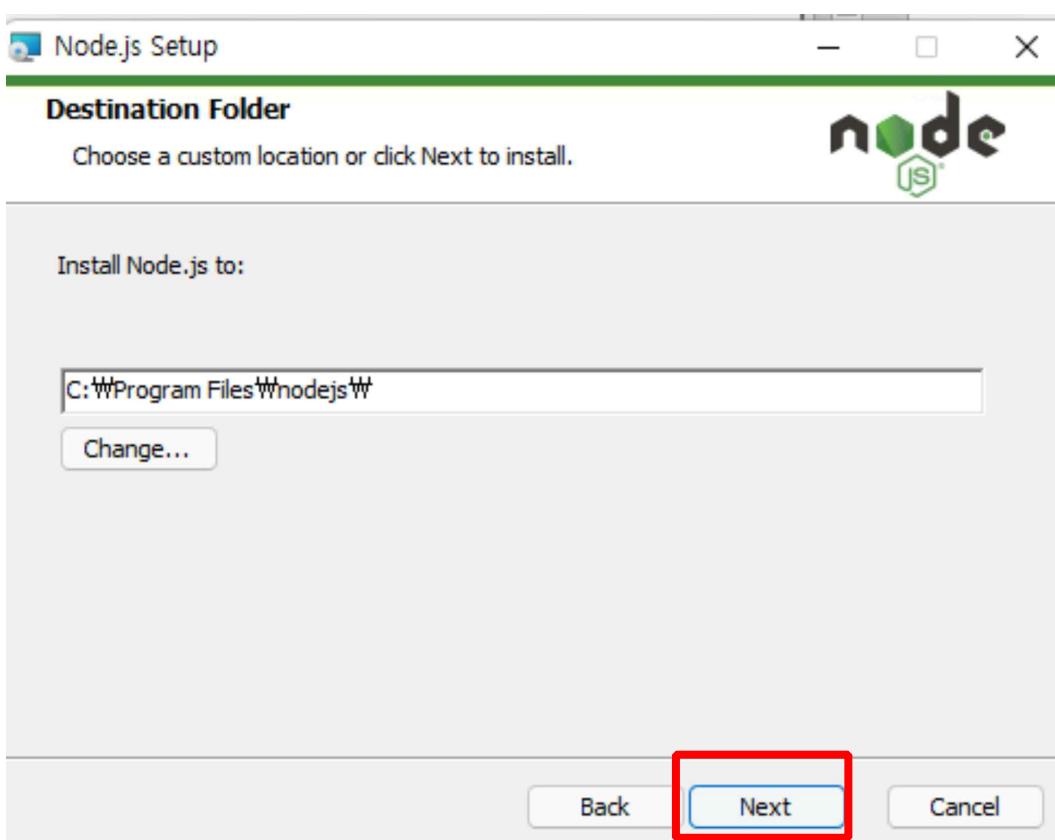
Download the latest version of nodeJS installation file from the link above.



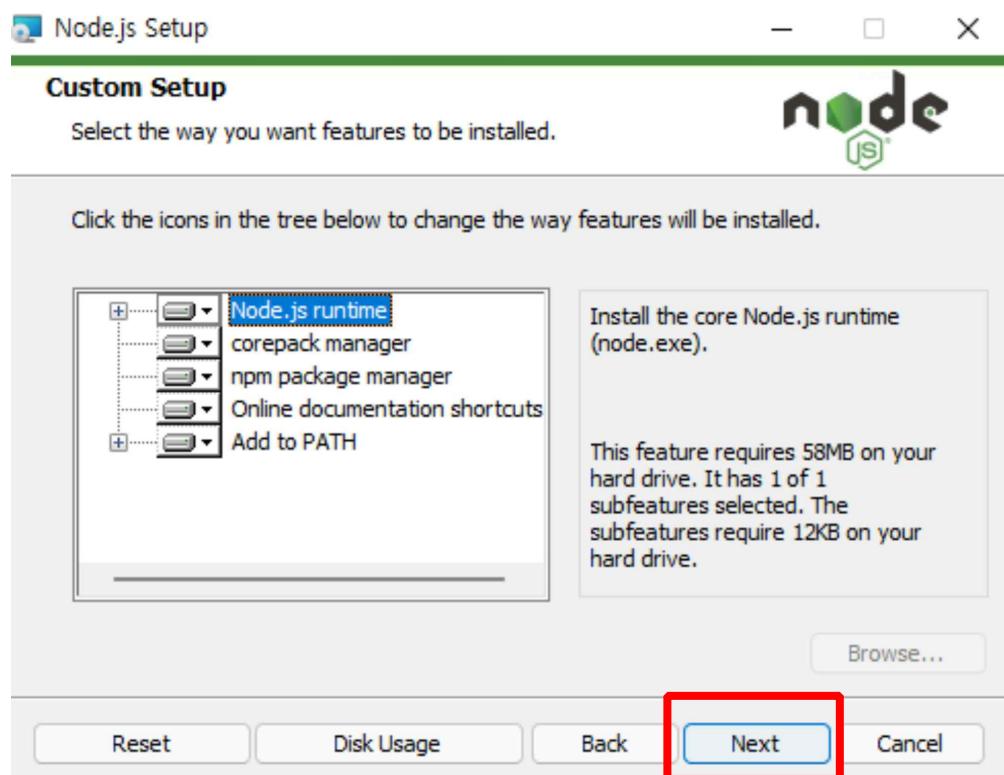
Next, run the downloaded installer as above. Click "Next".



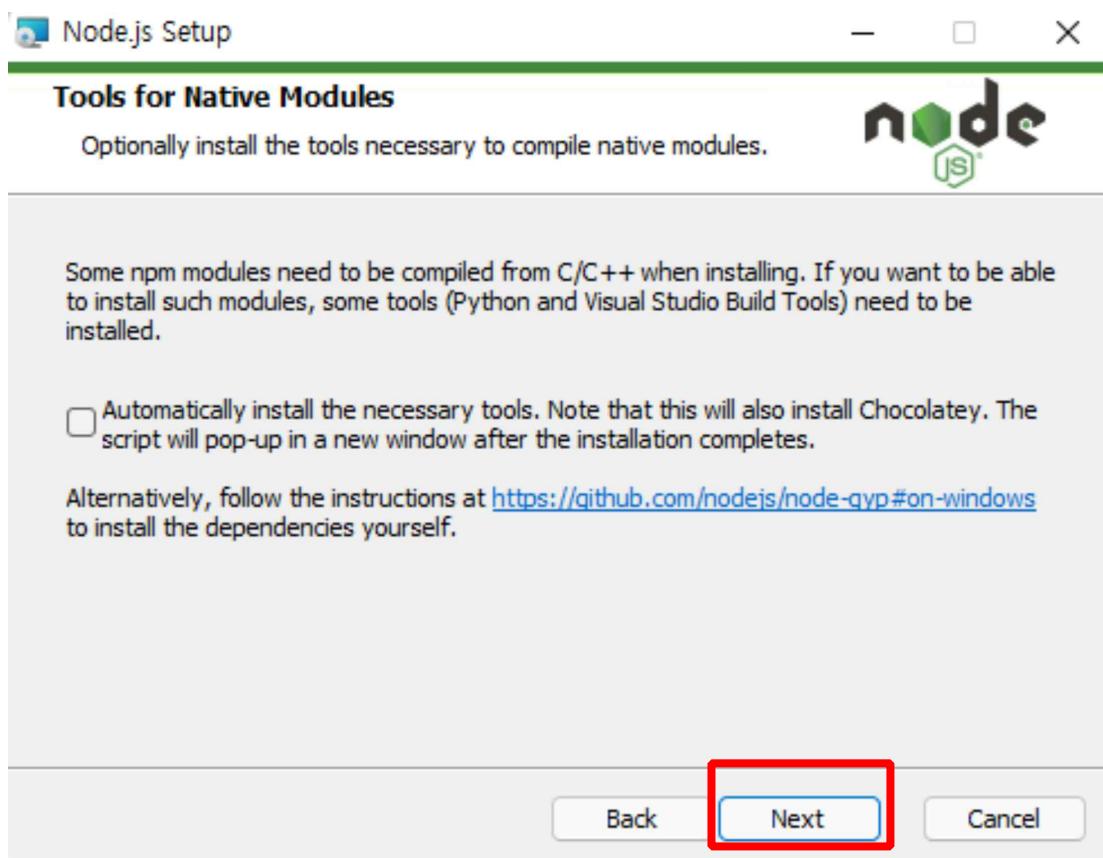
Accept the license for installation and click "Next".



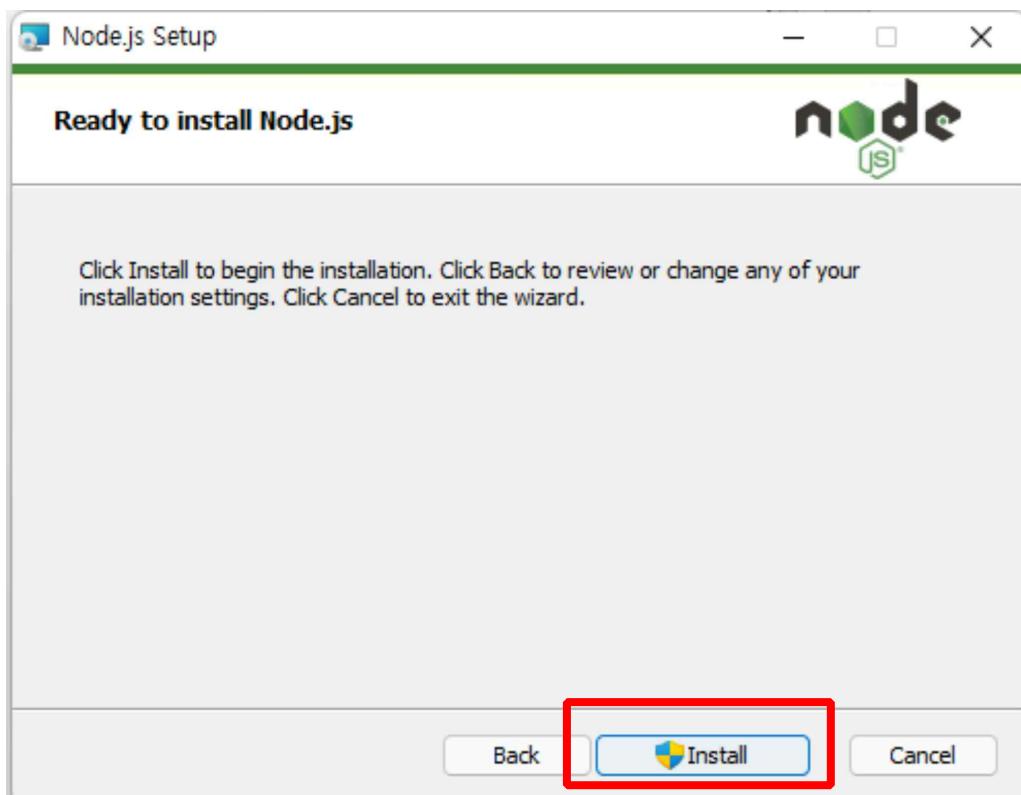
Click "Next".



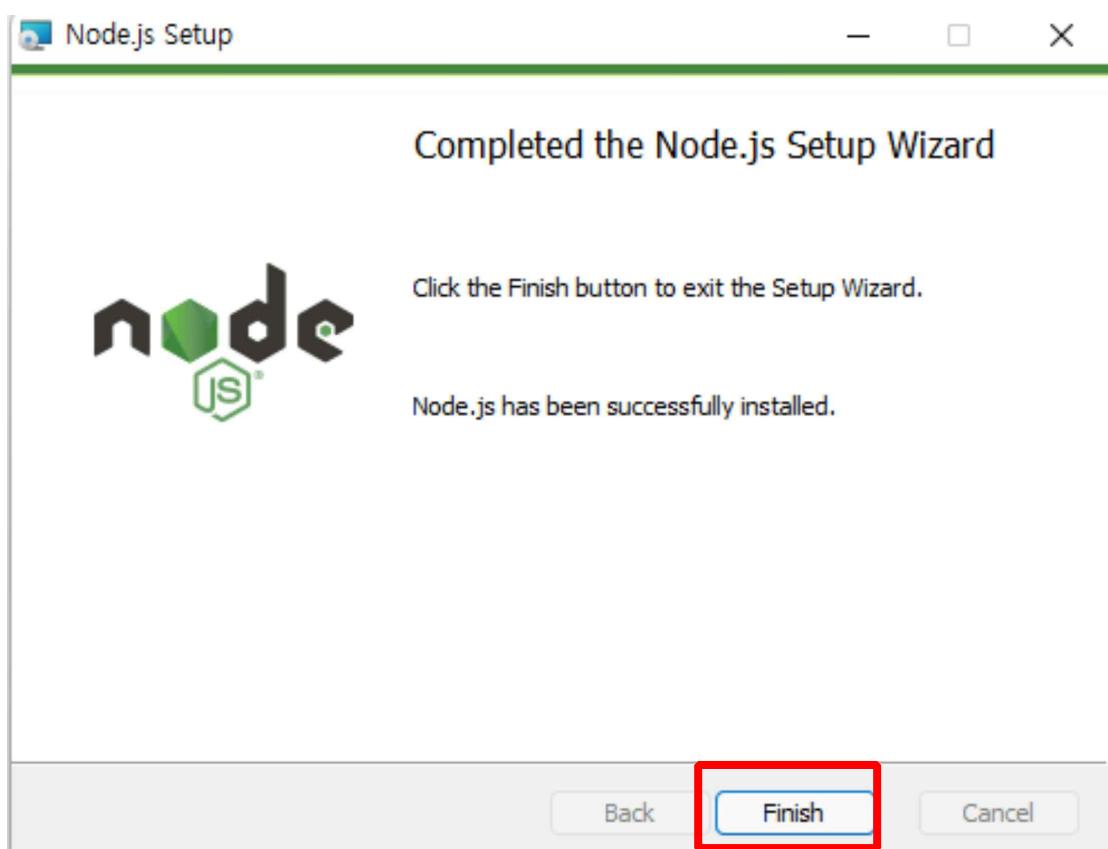
Click "Next".



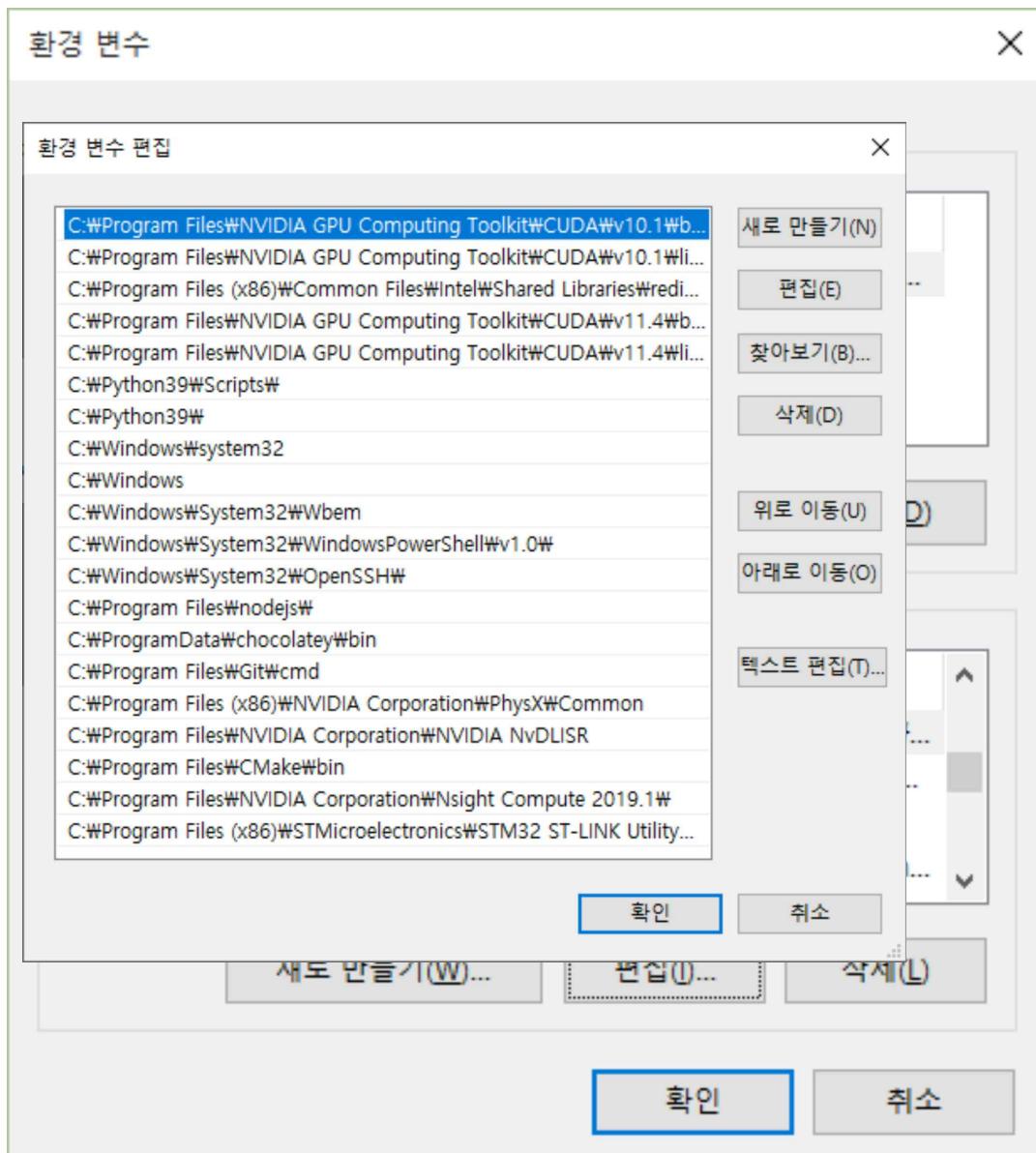
Click "Next".



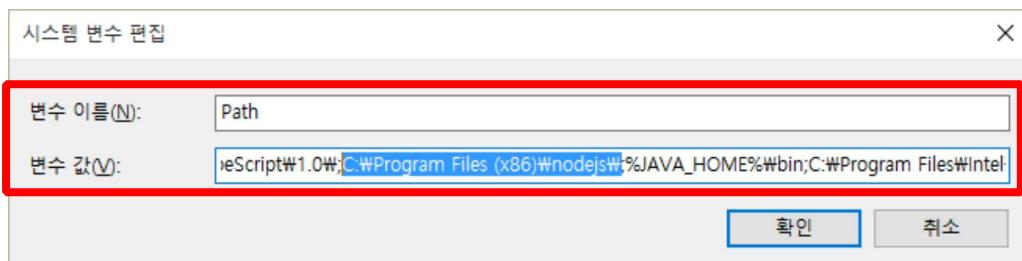
Click "Install" to begin installation.

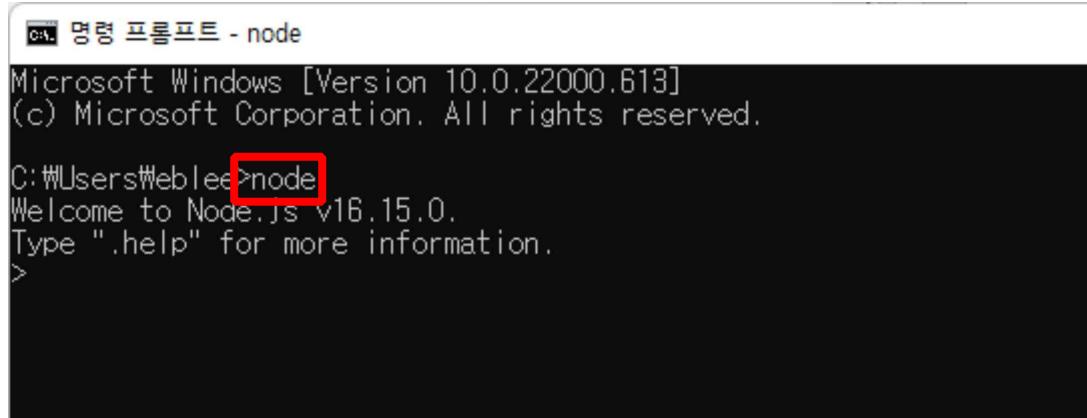


Click the "Finish" when the installation is complete.



In the case of NodeJs, the environment variable setting is automatically set during installation. After installation, check if NodeJS is added to the environment variable. If not, click "New" and add "C:\Program Files (x86)\nodejs\" as shown below and press the "OK" button.





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

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

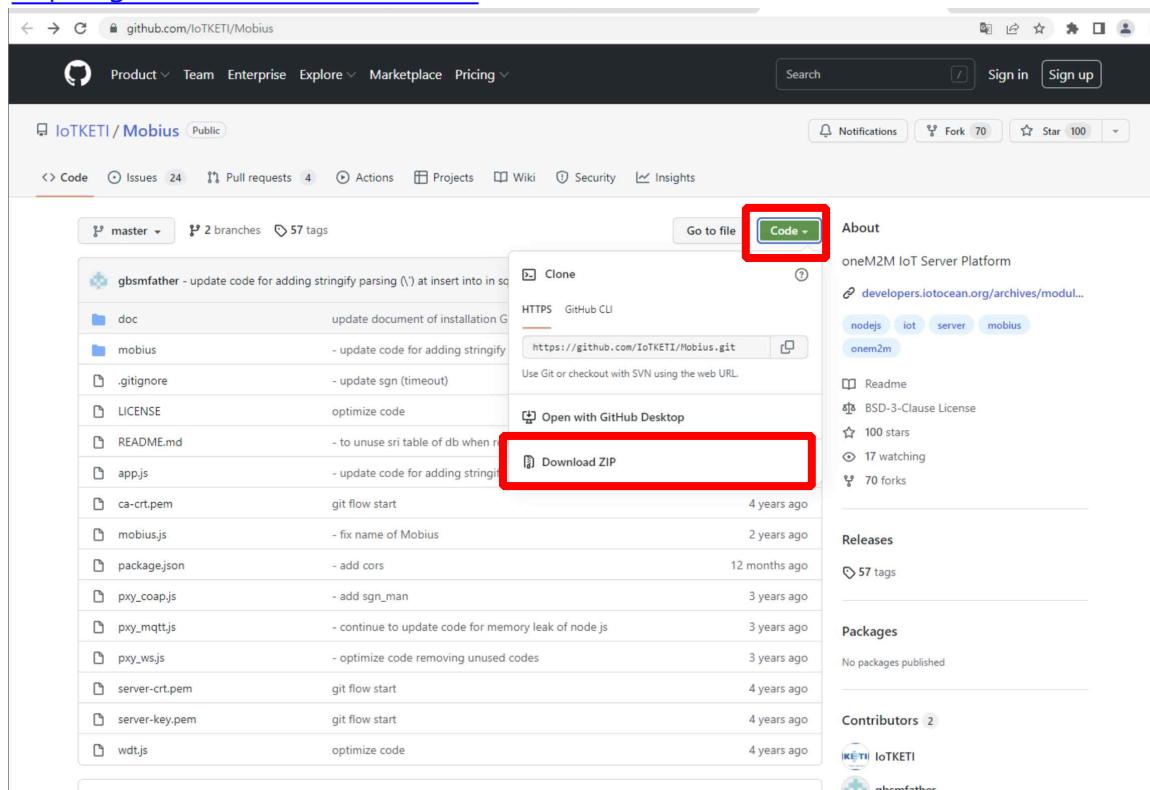
Verify that the node is properly installed. Node.js provides an interactive cli. If executed as above, node.js is properly installed.

2.3. Mobius Installation

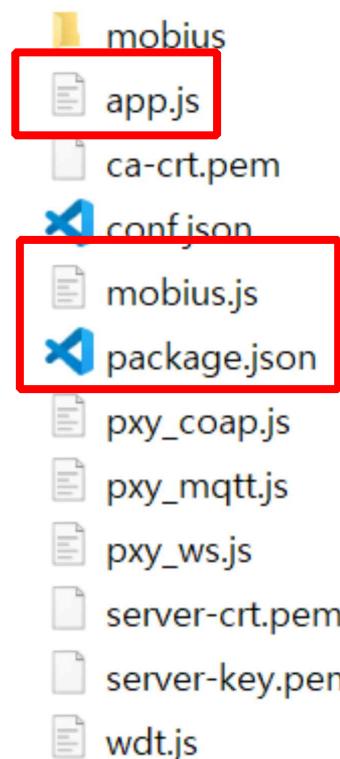
Mobius server platform source is available at GitHub. After downloading the Mobius server platform compression file, configuration information such as port information is set according to the provided manual.

2.3.1. Download and Module Installation

The Mobius installation file downloads the Mobius server file from the OCEAN Alliance at (<https://github.com/IoTKETI/Mobius>)



If you extract the downloaded file from the appropriate location, you can view the Mobius files of the node.js version as follows. Node.js does not require a separate compilation process, so it can be extracted and executed immediately. If the previous MySQL and Node.js are properly installed, installing the node.js module used in Mobius can run without any additional configuration.



A file in which mobius.js executes the main code: app.js that performs the gate and routing roles. Files in the "mobius" folder are codes that process actual packets and perform DB access and response.

Currently, additional node.js modules are not installed, which will cause an error. The command prompt should be executed as follows and the necessary modules should be installed using the npm tool. The package.json file has a list of required modules.

```
C:\Users\blee\Desktop\#cd Mobius-master
C:\Users\blee\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

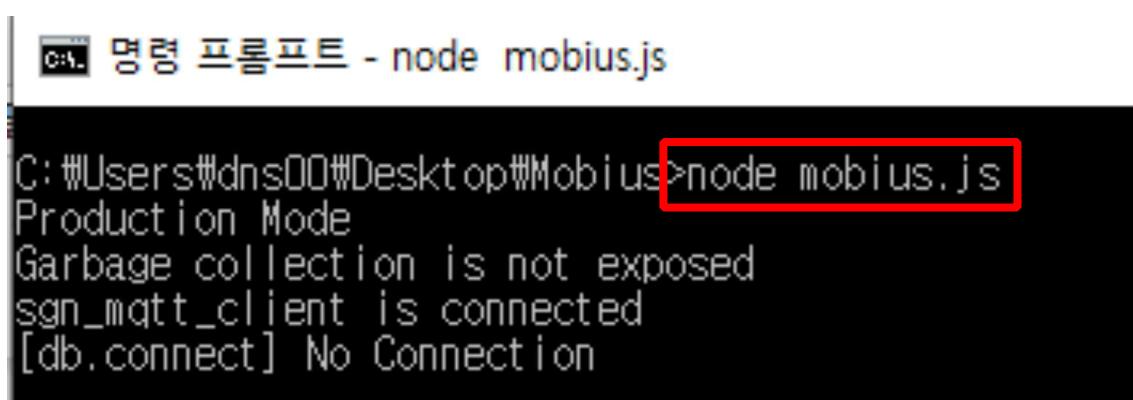
To address all issues (including breaking changes), run:
  npm audit fix --force

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\blee\Desktop\#Mobius-master>
```

If you enter the npm install command in the downloaded Mobius folder as above, the necessary modules are installed.

	mobius	Once installed, node_modules are shown on the left
	node_modules	You can see that the folder has been created.
	app.js	
	ca-crt.pem	
	conf.json	
	mobius.js	
	package.json	
	package-lock.json	
	pxy_coap.js	
	pxy_mqtt.js	
	pxy_ws.js	
	server-crt.pem	
	server-key.pem	
	wdt.js	



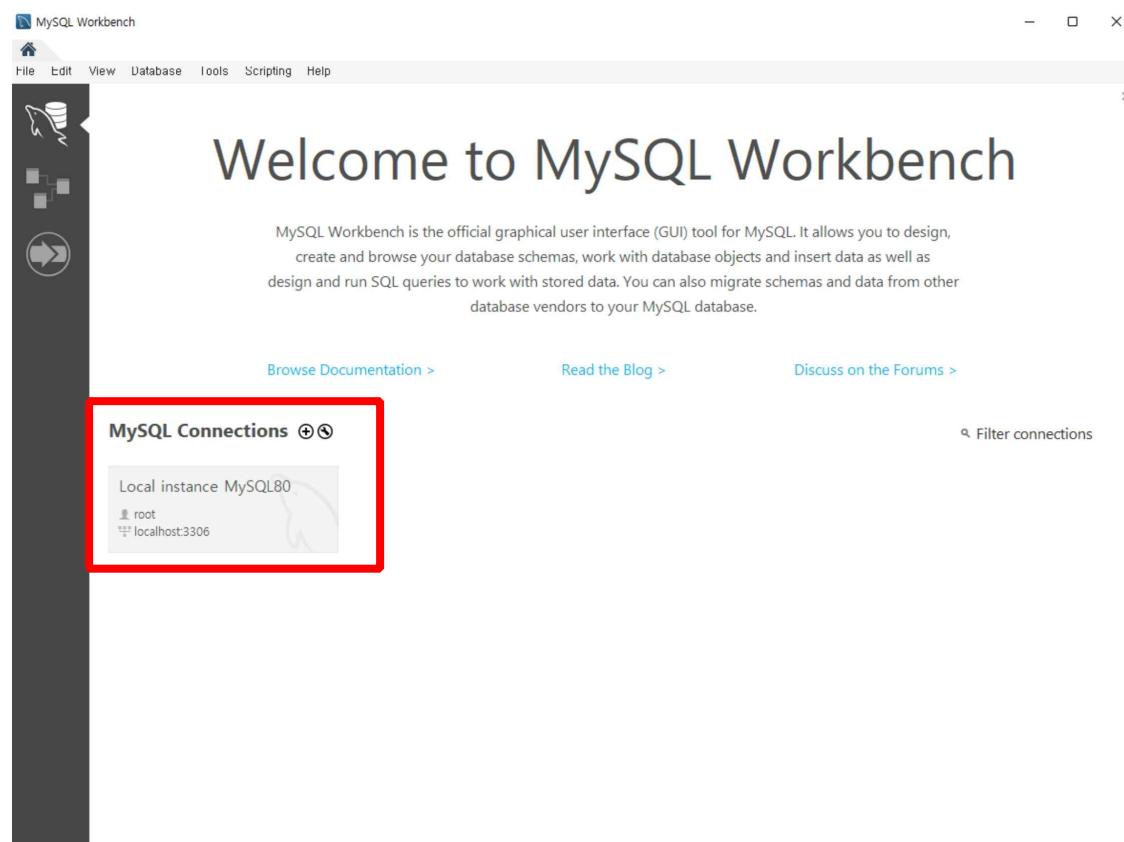
```
C:\Users\dns00\Desktop\Mobius>node mobius.js
Production Mode
Garbage collection is not exposed
sgn_mqtt_client is connected
[db.connect] No Connection
```

The installation of the module to drive Mobius is now complete.

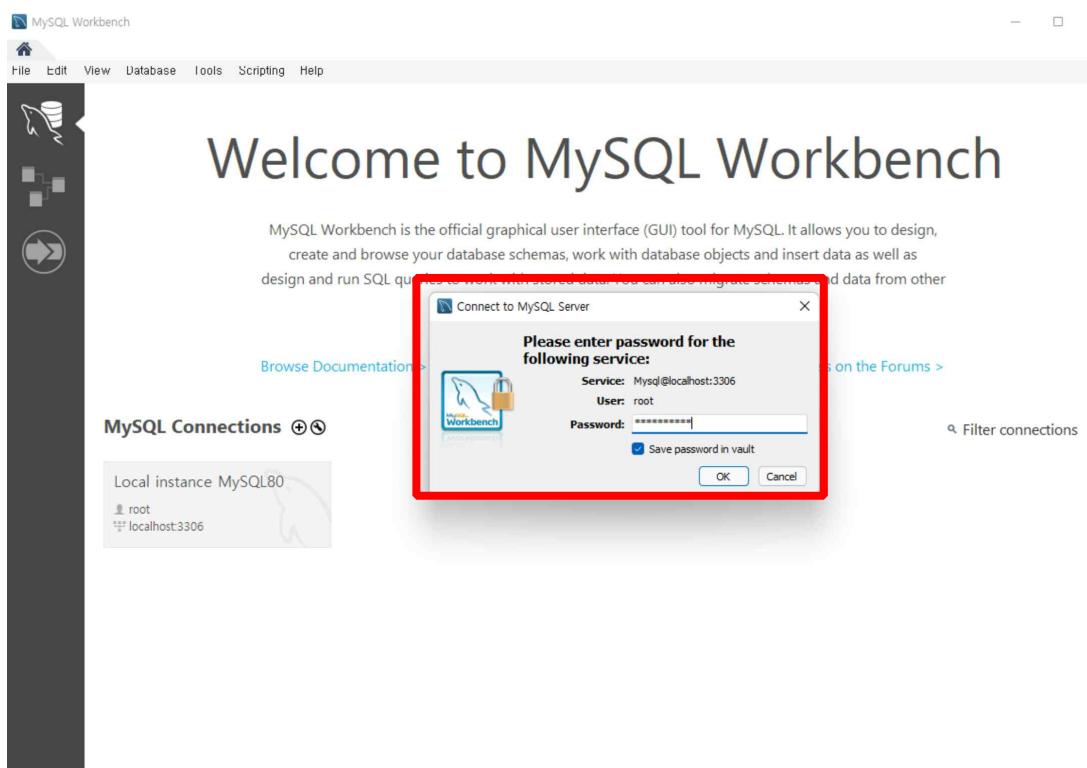
However, upon executing the command: "node mobius.js" as shown in the figure above, if MySQL and the database are not linked, an error occurs. Therefore, a process of linking Mobius and MySQL is necessary.

2.3.2. mobiusdb Database Creation

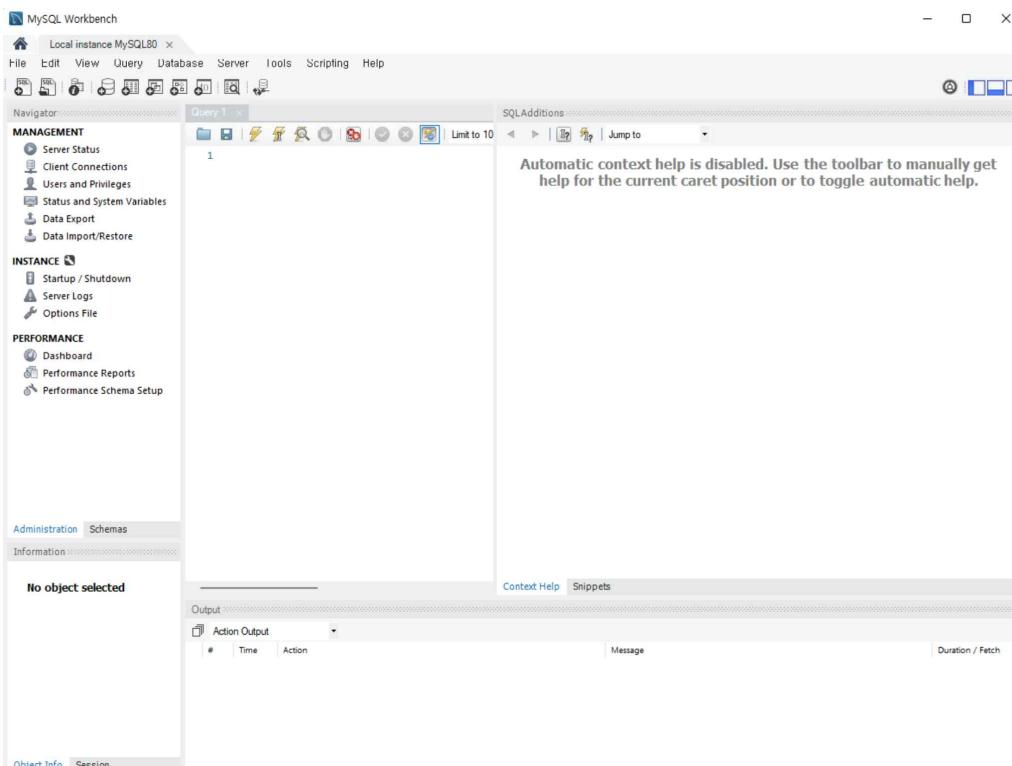
Since MySQL is installed in the Mobius-driven development environment above, MySQL Workbench should be installed.



Run MySQL Workbench and select Connections.

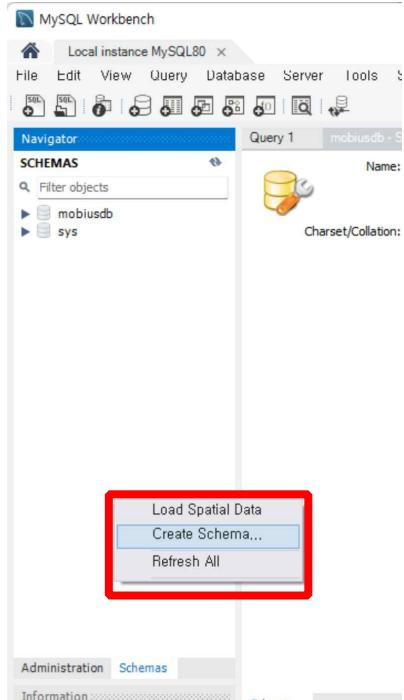


Enter the password you set when you installed MySQL.

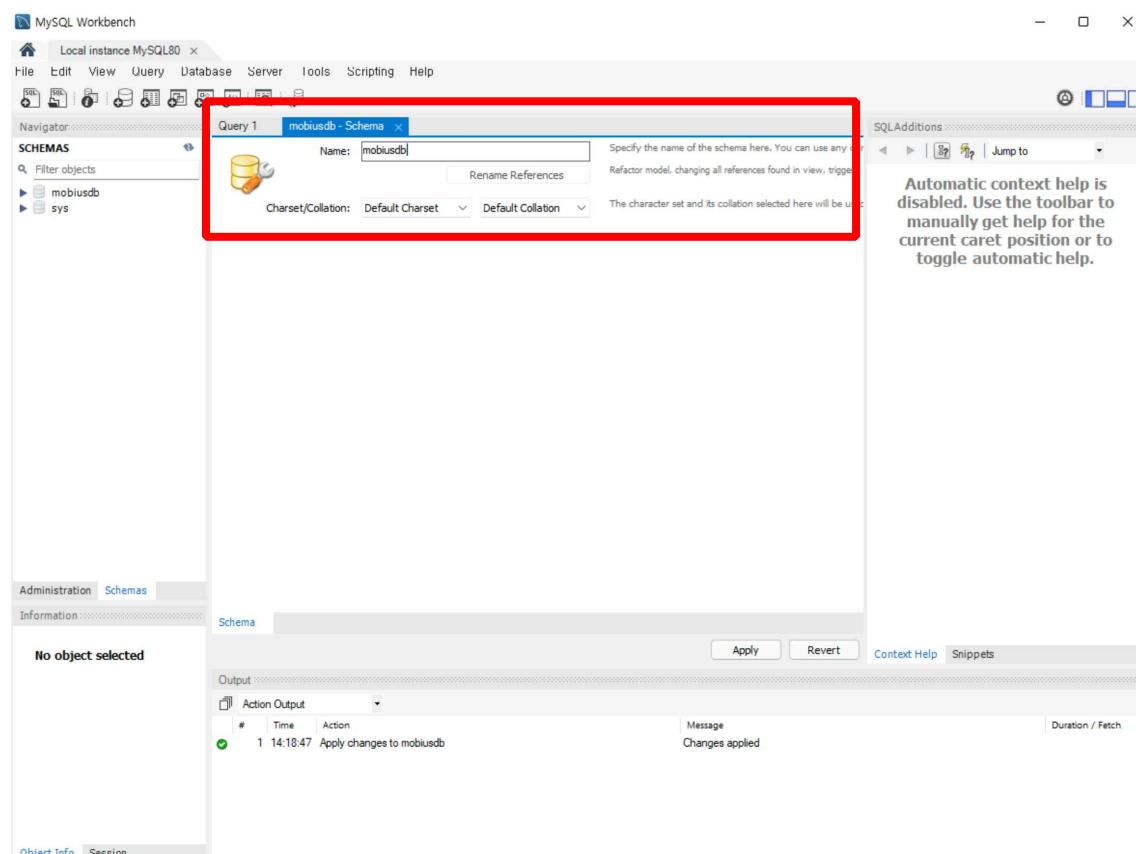


Workbench is a program that can handle MySQL in a GUI environment that can perform SQL queries, as well as various MySQL management tasks such as database, table creation, import, and export.

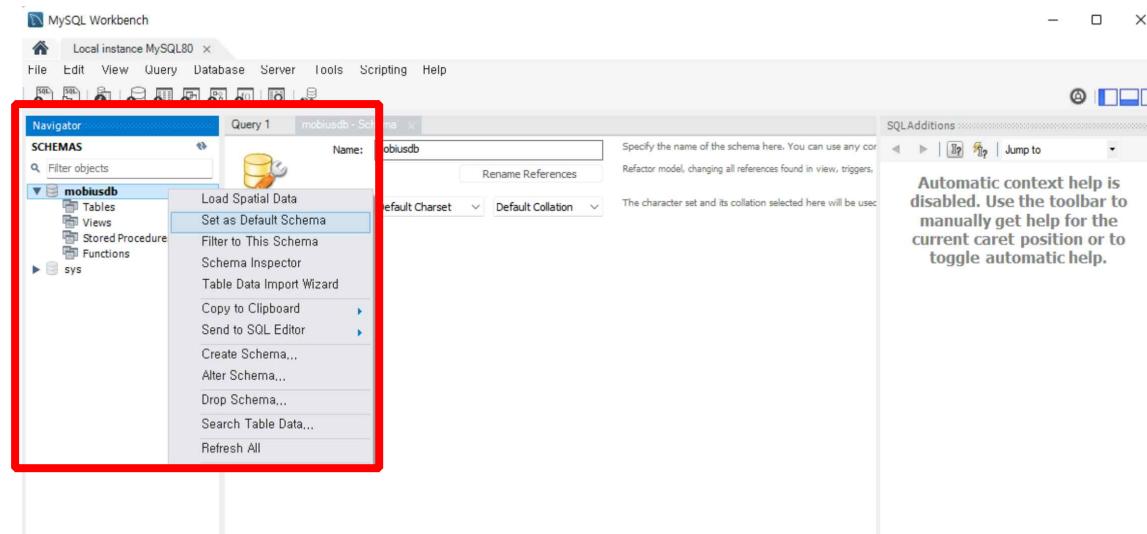
- Database Creation



Select “**Create Schema...**” from the menu that appears when you right-click on the SCHEMAS blank as shown above



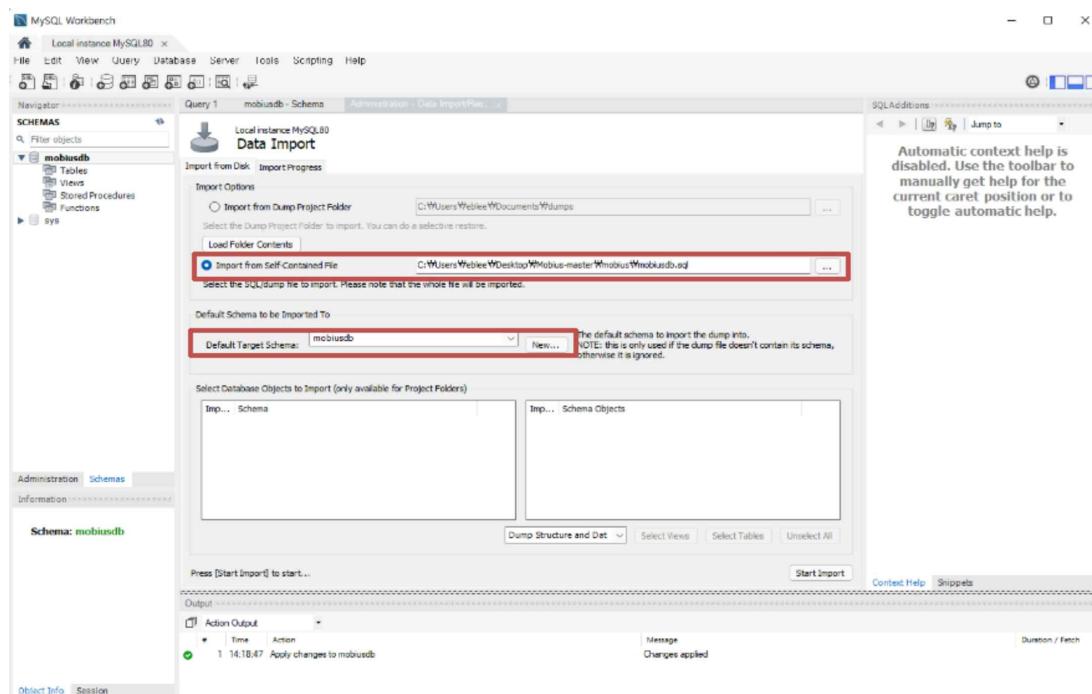
When the Schema Creation window appears, enter the database name "**mobiusdb**" and press the Apply button to create the database.



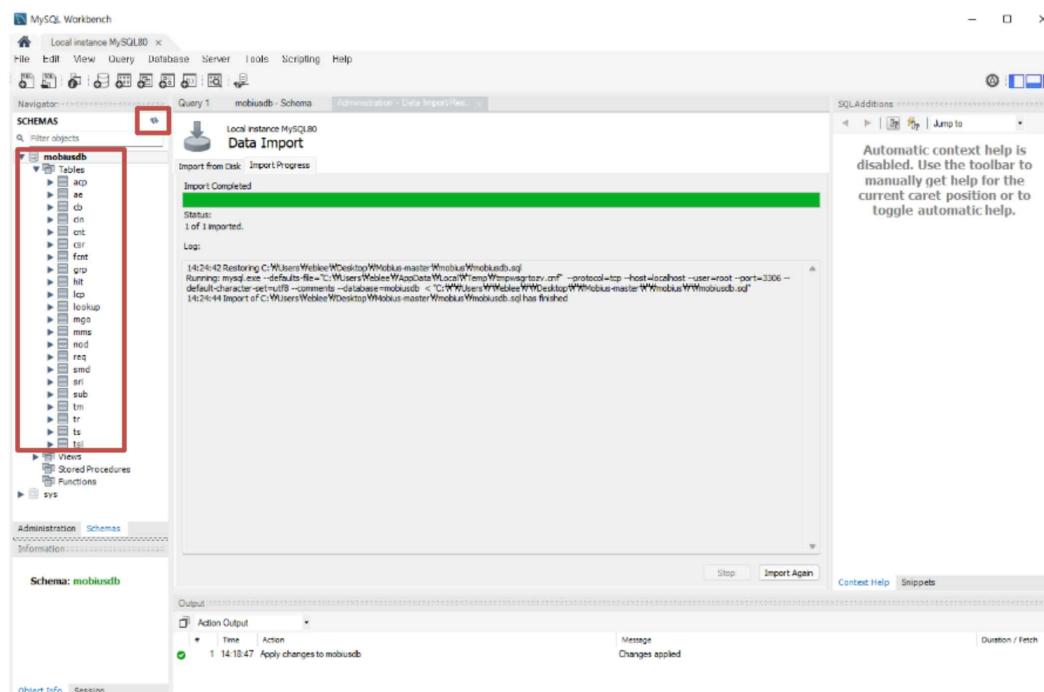
Once the "mobiusdb" data is generated, you can see that the database is created in the left pane as shown in the above picture, and set it to **Set as Default Schema** by right-clicking it.

- Import tables to mobiusdb

- 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



Select Data Import from the Server menu on the Workbench to import the table into the mobiusdb. In the Data Import window, select Import from Self-Contained File and select the mobiusdb.sql file in the figure above. Then, select the previously created mobiusdb with the Default Target Schema and press “Start Import” to create a table used in Mobius as shown in the following figure.



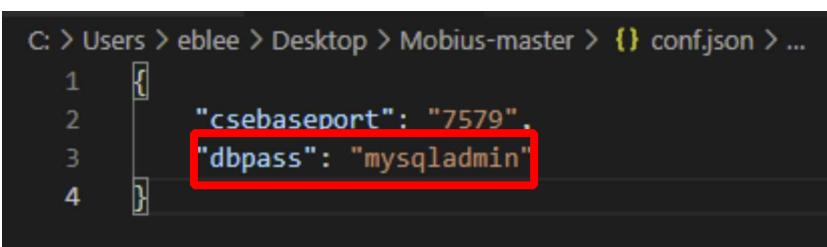
If you do not see the table after performing the data import, you must click the “refresh” icon above to view the imported tables.

3. Mobius Server Platform Drive

After installing the Mobius server normally, the environment must be set before the Mobius server can be run normally.

3.1. Environment Setting

Mobius configuration settings can be changed through the conf.json file.

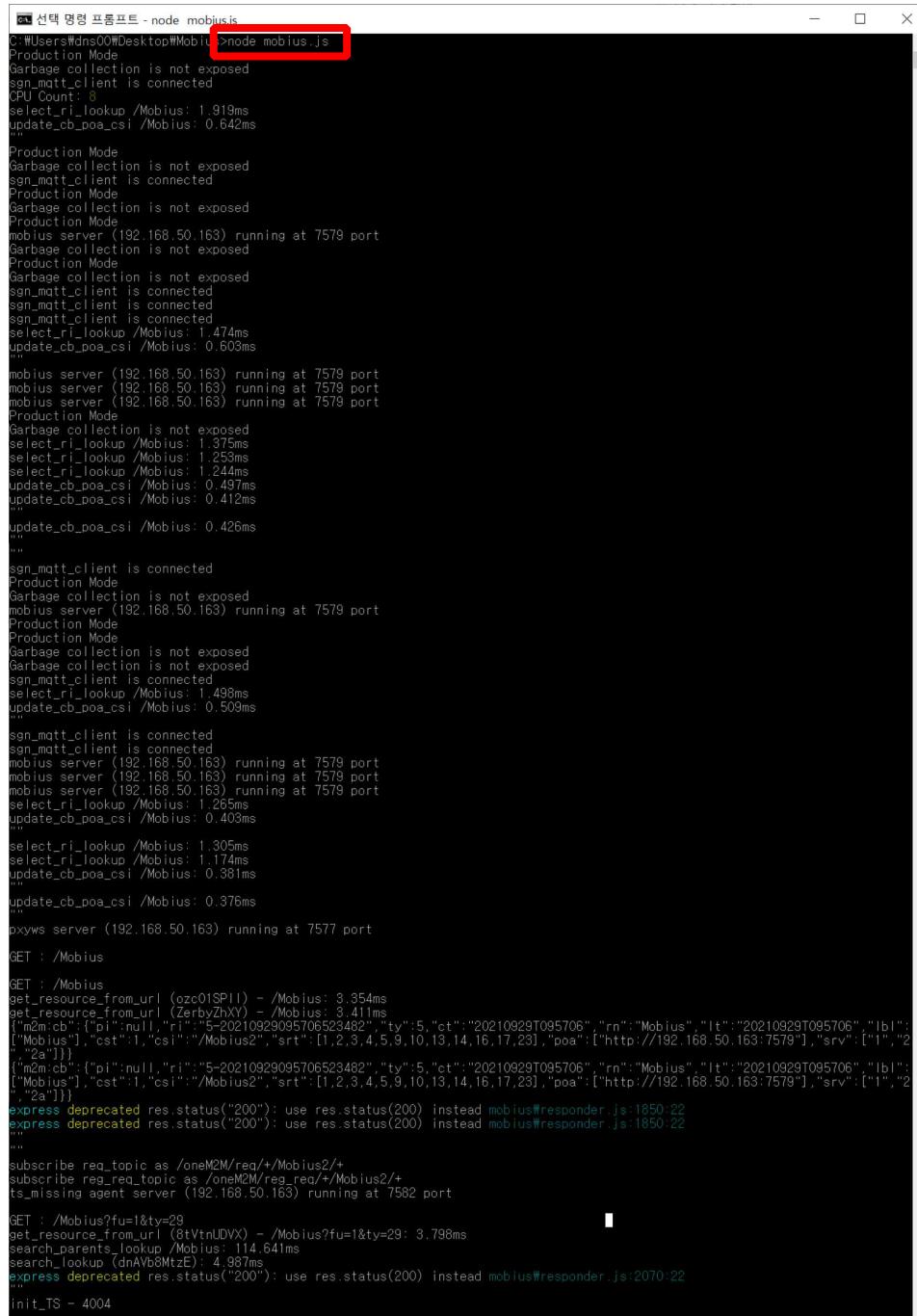


```
C: > Users > eblee > Desktop > Mobius-master > {} conf.json > ...
1  {
2    "csebaseport": "7579",
3    "dbpass": "mysqladmin"
4 }
```

The configuration variables are required to be set as shown in the figure. The port number "csebaseport" is required from where the Mobius server platform listens to the requests. The database password "dbpass" is required for the Mobius to access the MySQL database.

After updating the configuration information, you can run the 'node mobius.js' command in the installed folder as follows to run the Mobius server platform.

If you want to change additional settings, you have to change the source code directly. Additional configuration changes are made through the mobius.js file. The mobius.js file will look like this.



```

[1] 선택 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
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
Production Mode
Garbage collection is not exposed
Garbage collection is not exposed
sgn_mqtt_client is connected
select_ri_lookup /Mobius: 1.498ms
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
"""

pxys server (192.168.50.163) running at 7577 port
GET : /Mobius
GET : /Mobius
get_resource_from_url (ozc01SP1) - /Mobius: 3.354ms
get_resource_from_url (ZerbyZhXY) - /Mobius: 3.411ms
["m2n:cb": {"pi": null, "ri": "5-20210929095706523482", "ty": 5, "ct": "20210929T095706", "rn": "Mobius", "lt": "20210929T095706", "lb": "Mobius"], "cst": "Mobius2", "srt": [1, 2, 3, 4, 5, 9, 10, 13, 14, 16, 17, 23], "poa": ["http://192.168.50.163:7579"], "srv": ["1", "2", "23"]}
["m2n:cb": {"pi": null, "ri": "5-20210929095706523482", "ty": 5, "ct": "20210929T095706", "rn": "Mobius", "lt": "20210929T095706", "lb": "Mobius"], "cst": "1", "csf": "Mobius2", "srt": [1, 2, 3, 4, 5, 9, 10, 13, 14, 16, 17, 23], "poa": ["http://192.168.50.163:7579"], "srv": ["1", "2", "23"]}
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/reg/+/Mobius2/
subscribe req_req_topic as /oneM2M/reg_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 (dnAVb8MtZc): 4.987ms
express deprecated res.status('200'): use res.status(200) instead mobius#responder.js:2070:22
init_TS - 4004

```

Setting the information on the mobius.js server

- usecsetype: The server type, Mobius, must be entered as 'in' as IN-CSE.
- usecsebase: Indicates the CSEBase name for Mobius. By default, it is mobile-yt.
- usecseid: Enter the cseid for Mobius. The cseid must begin with '/' and, by default, '/' is used before the csebase name.
- usedbhost: DB server address, using localhost by default.
- usemqttbroker: mqtt broker host address, using localhost by default.

3.2. Test

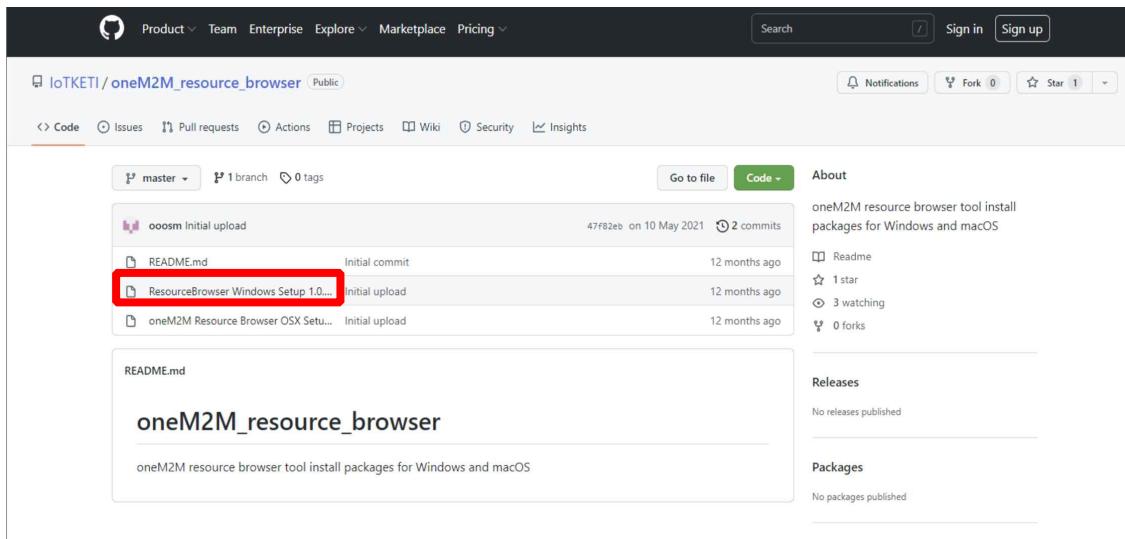
3.2.1. Resource Viewer(oneM2M Browser) Application

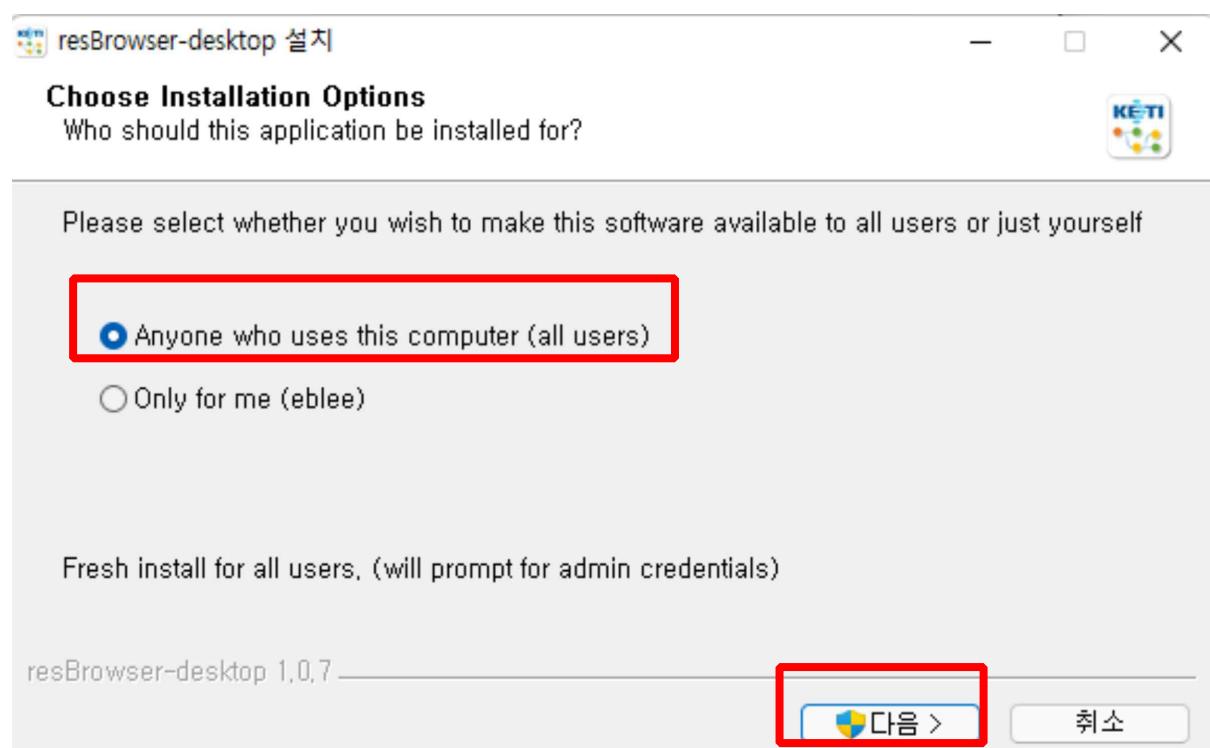
You can use the Web-based resource viewer to test whether devices created through Mobius and nCube work properly. Similar to Windows File Explorer, the Resource Viewer can display resources in a tree structure and perform simple actions of resource creation and deletion.

The resource viewer of the web version is provided in the url below.

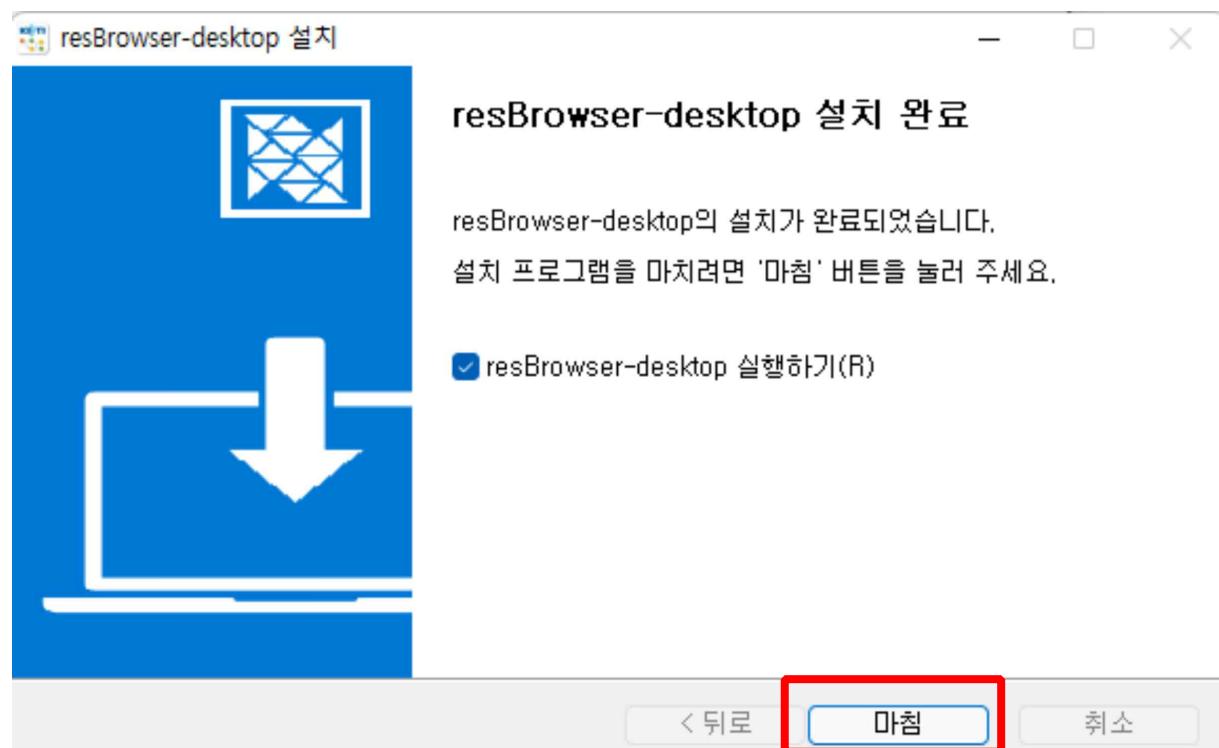
<http://203.253.128.161:7575>

A tool that allows you to view resources only from the web version of the resource viewer, and a version that allows you to view the structure of other oneM2M platforms is provided by OCEAN's GitHub as the oneM2M Browser of the Windows application version.

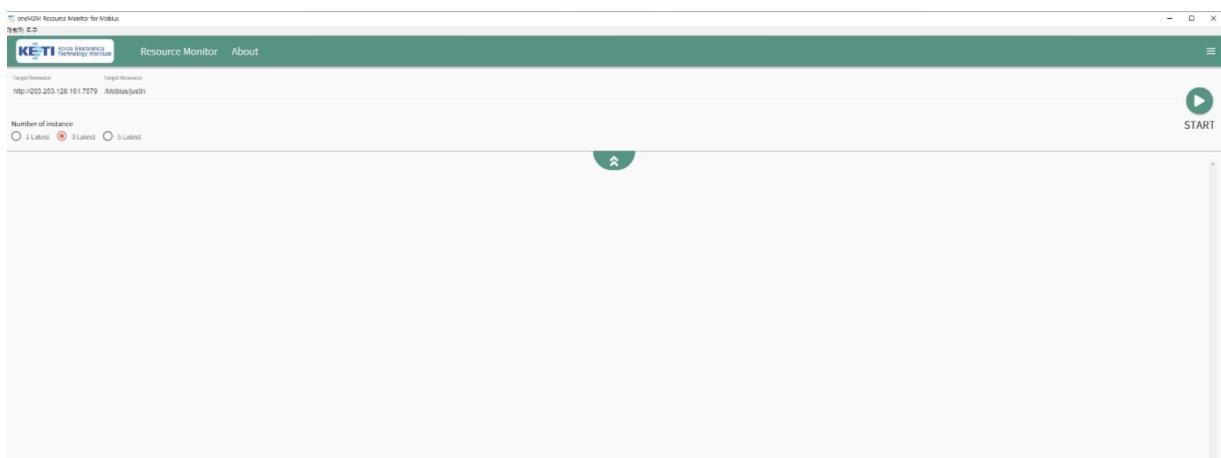




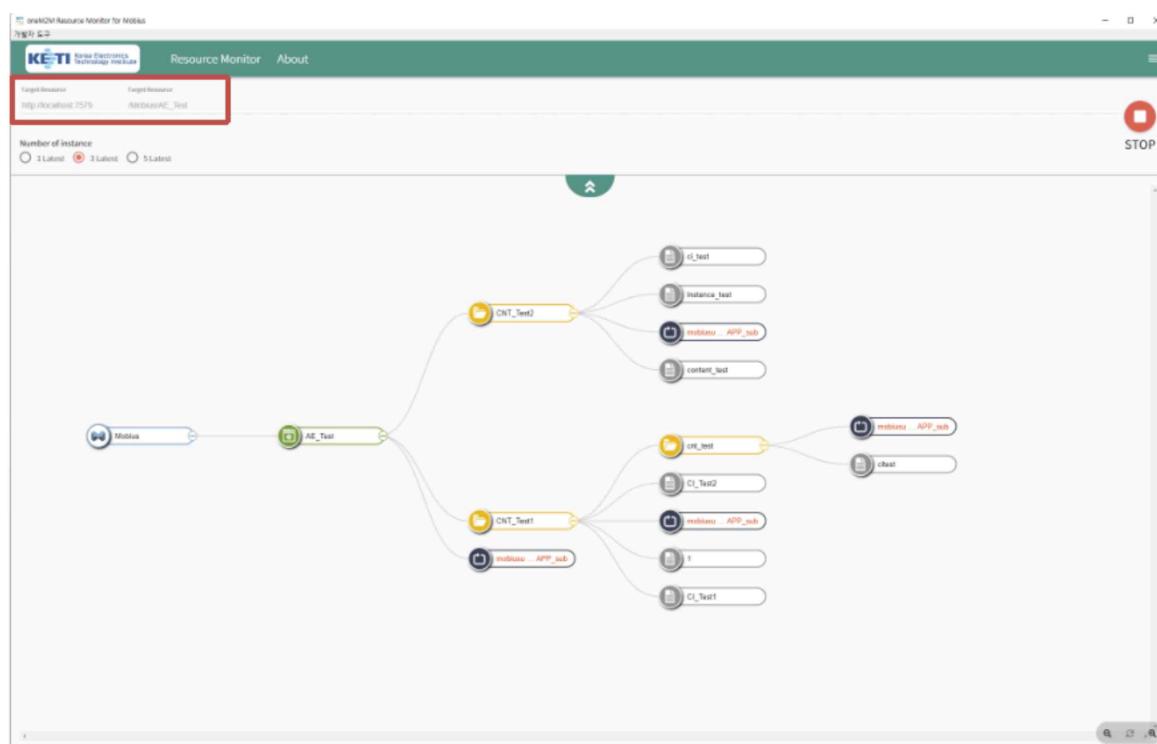
Click "Next" to install the Resource Viewer.



Click "Finish" when the installation is complete.



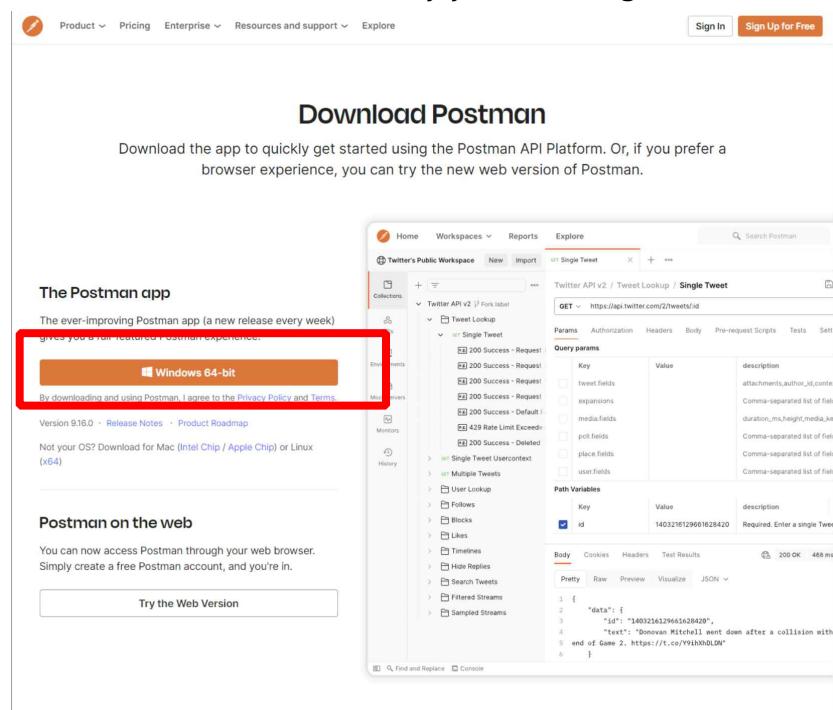
When the resource viewer installation is completed, it is executed as above.



When the resource viewer is visible as shown below, enter the url of the resource created in Mobius in the displayed part and press the "Start" button to view the resource structure of the url. The first part of the url is a combo box, and you can select the previous Mobius and Mobius release 2. After that, you must enter the name of the AE you created here.

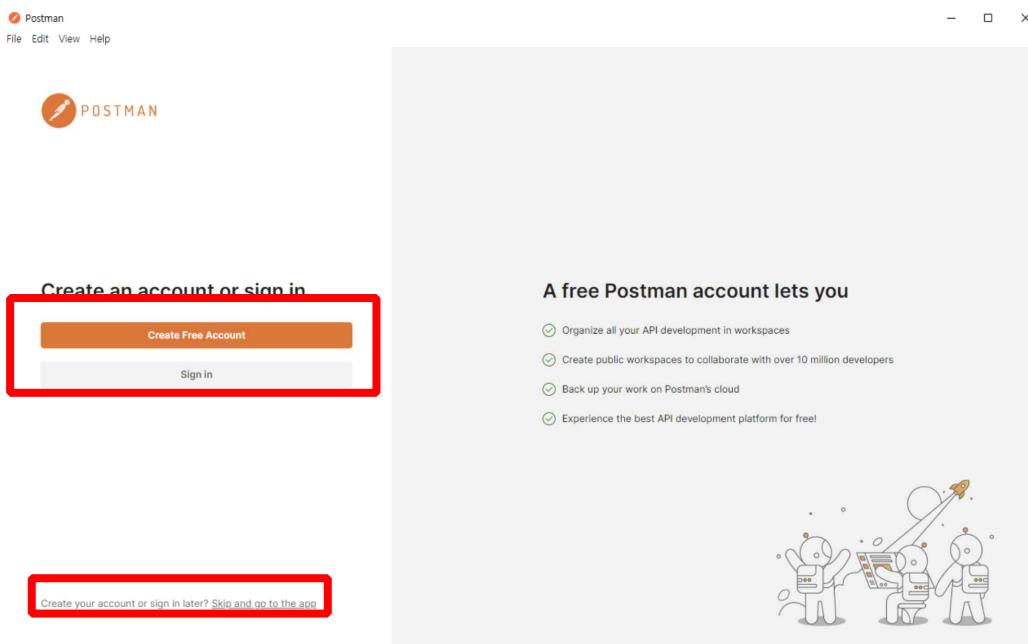
3.2.2. Postman Application

You can test the behavior of Mobius built by yourself using Postman.



The screenshot shows the Postman download page. At the top, there's a navigation bar with links for Product, Pricing, Enterprise, Resources and support, and Explore. On the right, there are Sign In and Sign Up for Free buttons. Below the navigation, a large orange button says "Download Postman". Underneath it, text reads: "Download the app to quickly get started using the Postman API Platform. Or, if you prefer a browser experience, you can try the new web version of Postman." A section titled "The Postman app" shows a download button for "Windows 64-bit" which is highlighted with a red box. Below this, there's a note about the ever-improving Postman app and links for Release Notes and Product Roadmap. Another section titled "Postman on the web" has a "Try the Web Version" button. To the right, a screenshot of the Postman interface shows a request for "Twitter API v2 / Tweet Lookup / Single Tweet" with various parameters like "id" and "expansions".

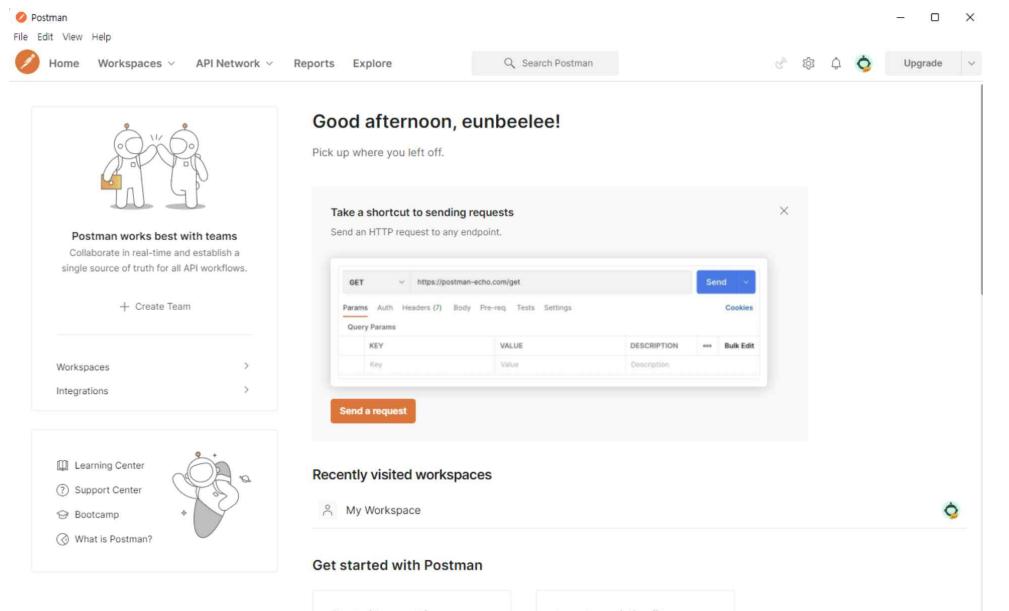
<https://www.postman.com/downloads/> You can access this site and download the postman.



The screenshot shows the Postman sign-in page. At the top, there's a menu with File, Edit, View, Help, and the Postman logo. The main area has two sections: "Create an account or sign in" on the left with a "Create Free Account" button highlighted with a red box, and "A free Postman account lets you" on the right with a list of benefits: Organize all your API development in workspaces, Create public workspaces to collaborate with over 10 million developers, Back up your work on Postman's cloud, and Experience the best API development platform for free! Below these sections, there's a link "Create your account or sign in later? Skip and go to the app." which is also highlighted with a red box.

After installing Postman, log in after signing up for membership.

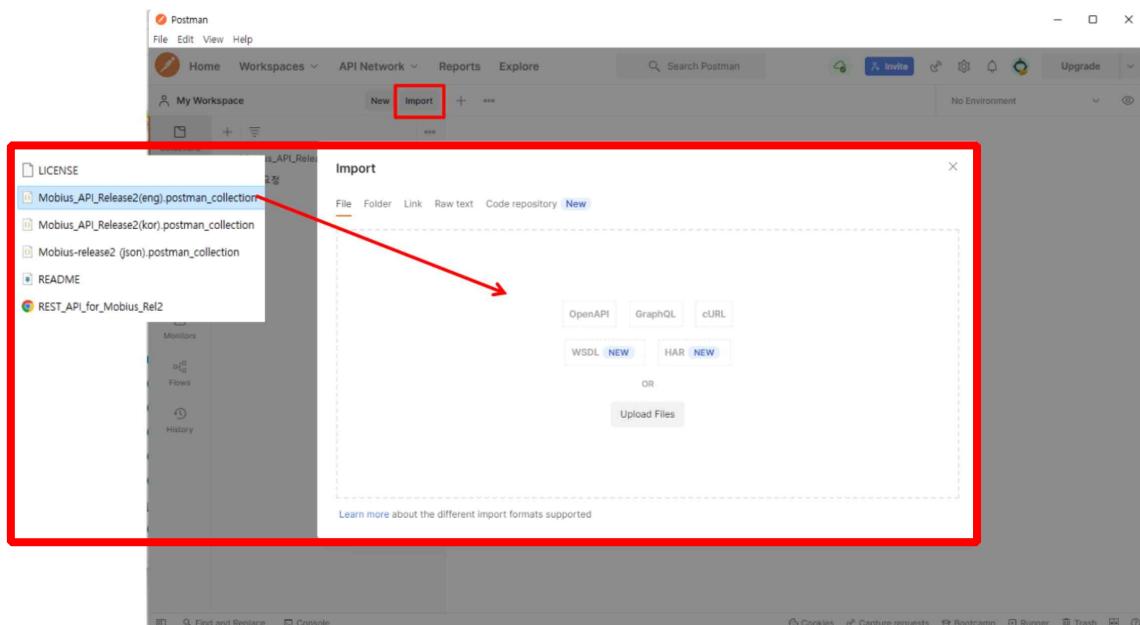
If you don't want to log in, skip below.



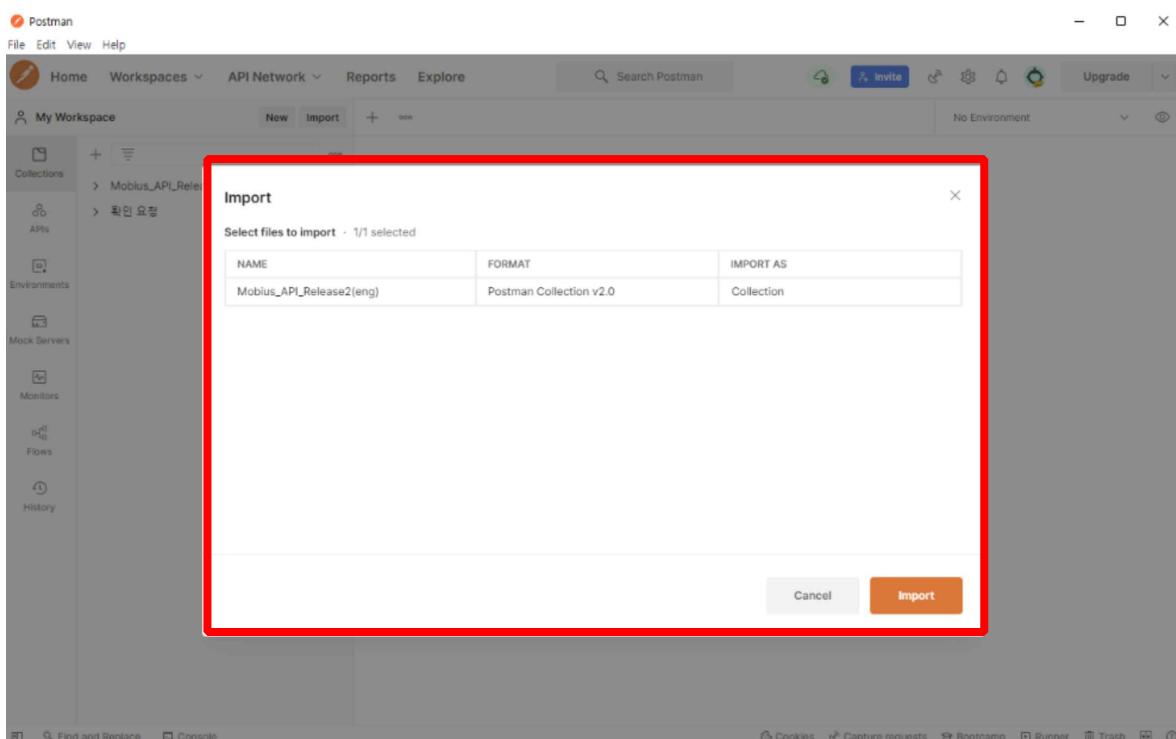
If you log in after installing Postman, it will run as above.

The screenshot shows a GitHub repository page for 'IoTKETI / oneM2M-API-Testing'. The page includes a navigation bar with 'Product', 'Team', 'Enterprise', 'Explore', 'Marketplace', 'Pricing', 'Search', 'Sign in', and 'Sign up'. The repository details show 1 issue, 1 pull request, 0 actions, 0 projects, 0 wiki, 0 security, and 0 insights. The 'Code' tab is selected, with a red box highlighting the 'Code' button. The repository content lists files like 'LICENSE', 'Mobius-release2 (json).postman_col...', 'Mobius_API_Release2(eng).postman...', 'Mobius_API_Release2(kor).postman...', 'README.md', and 'REST_API_for_Mobius_Rel2.pdf'. On the right, there's an 'About' section with a 'Clone' button, a 'Go to file' button (also highlighted with a red box), and a 'Download ZIP' button (also highlighted with a red box). Other sections include 'Readme', 'BSD-3-Clause License', '6 stars', '2 watching', '10 forks', 'Releases' (1 release), and 'Packages' (no packages published).

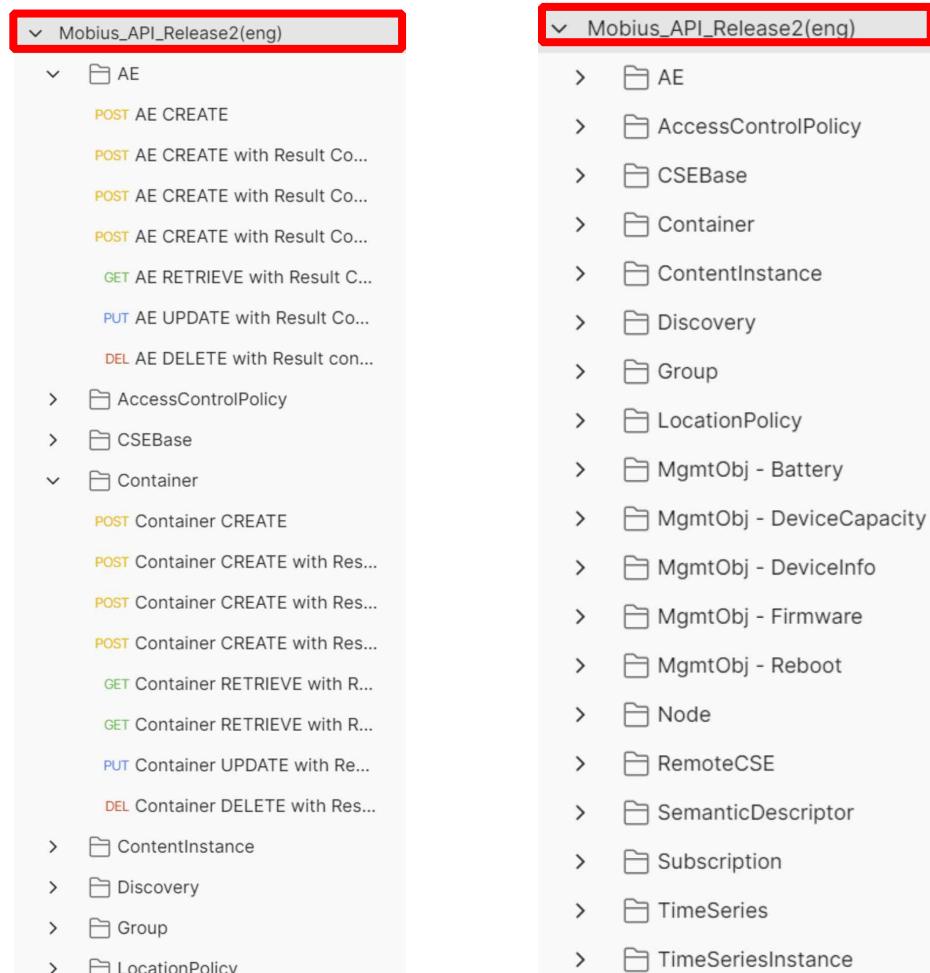
Download pre-written scripts from OCEAN for Mobius testing.



To import the scripts, select the "Import" button next to My Workspace as shown in the figure. Drag the script file downloaded from OCEAN when a window for import opens.



Click the "Import" button to complete the import.



If the import succeeds, a collection tab called `Mobius_API_Release2(eng)` will be created as shown in the figure above. This test script is organized to allow CRUD of currently supported resources and is set to be tested in xml data format.



To test, you must enter the IP of the server where Mobius is running.

The value should be set in the address window as shown in the picture above as `{{mp_url}},{{cb}}`. Instead, you can use it by entering `http://(directly built server address):port number/csebase` value.

The screenshot shows the Postman interface. At the top, there's a search bar and various icons. Below it, a dropdown menu labeled 'No Environment' is highlighted with a red box. A red arrow points from this dropdown down to a second dropdown in a modal window titled 'Environment'. This second dropdown also has 'No Environment' selected. Another red arrow points from the 'Add' button at the bottom right of this modal window.

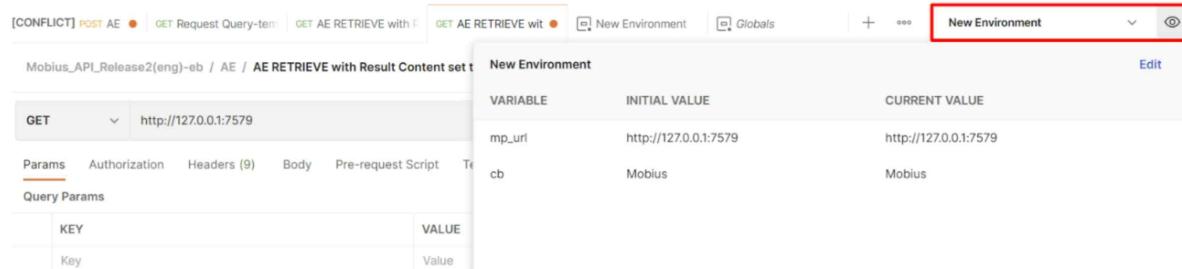
The method for setting `{{mp_url}}` is shown as at the top right in the above figure, there is a list box where you can set the environment and select it. On the Environment tab, select "Add".

The screenshot shows the 'New Environment' setup window. It has a table with two rows. The first row has 'VARIABLE' column containing 'mp_url' with a checked checkbox, and 'INITIAL VALUE' column containing 'http://127.0.0.1:7579'. The second row has 'VARIABLE' column containing 'cb' with a checked checkbox, and 'INITIAL VALUE' column containing 'Mobius'. There is a red box around the 'mp_url' variable row. A red box also surrounds the 'Save' button at the top right of the window.

When the setup window appears, `{{mp_url}}`, `{{cb}}` Enter a value.

- ① In VARIABLE name, enter mp_url, cb.
- ② Enter `http://127.0.0.1:7579` in INITIAL VALUE of mp_url. CURRENT VALUE is automatically entered.
- ③ Enter Mobius in INITIAL VALUE of cb. CURRENT VALUE is automatically entered.

When the setup is complete, press the Save button to add and close the window.



The screenshot shows the Postman interface with a new environment named "New Environment" highlighted by a red box. The environment table lists variables: mp_url (initial value: http://127.0.0.1:7579, current value: http://127.0.0.1:7579) and cb (initial value: Mobius, current value: Mobius). Below the table, there is a "Query Params" section with columns for KEY and VALUE.

VARIABLE	INITIAL VALUE	CURRENT VALUE
mp_url	http://127.0.0.1:7579	http://127.0.0.1:7579
cb	Mobius	Mobius

KEY	VALUE
Key	Value

If you return to the Postman window and select the settings you just added in the list box, you can use the script that you distributed without having to enter an address every time thereafter.