



OCEAN

nCube:Thyme for Java

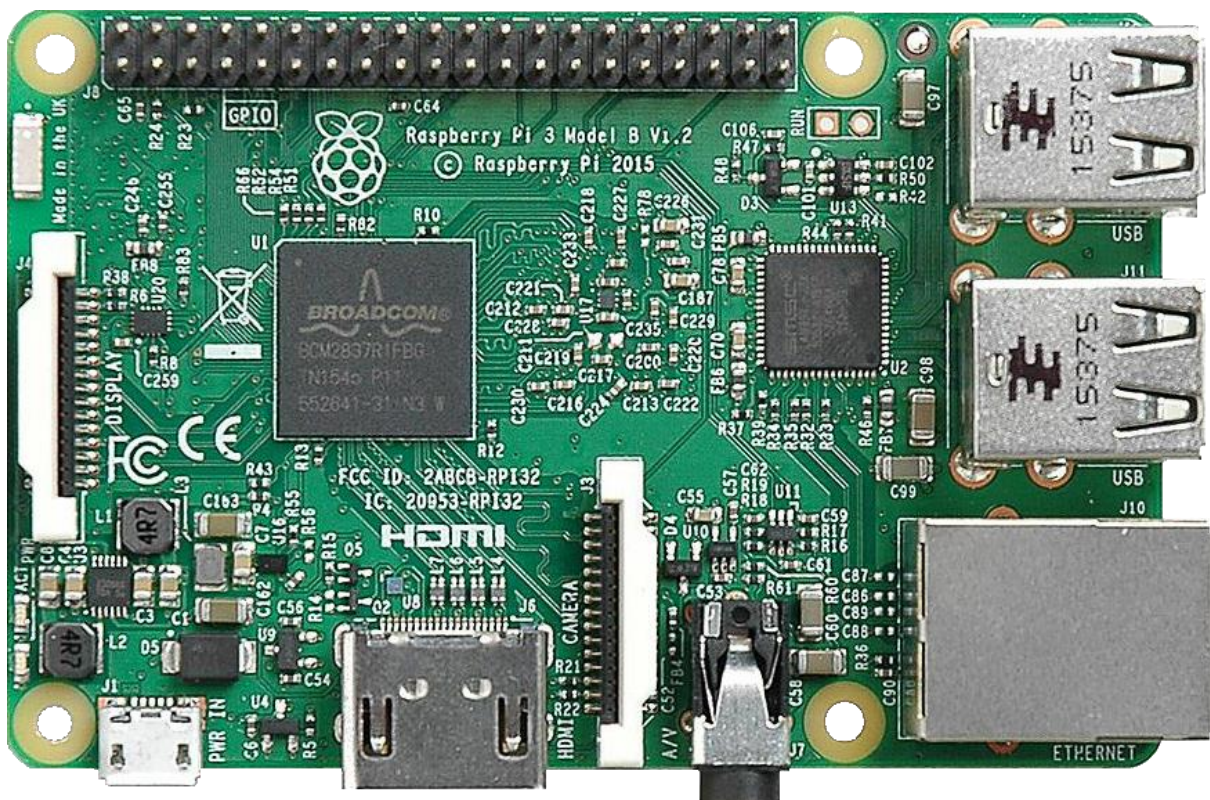
Ver.1.0

nCube:Thyme for Java

nCube:Thyme for Java 는 oneM2M AE Java 를 이용하여 구현한 것이다. 지원하는 통신 프로토콜은 MQTT, HTTP, CoAP 이다.

nCube:Thyme for Java 는 인터넷 연결과, Java 환경이 갖추어 진 어떠한 환경에서도 동작할 수 있으며, 본 문서에서는 Raspberry pi 3 Model B 를 기준으로 설명한다.

Mobius IoT Platform 에 생활속에서 수집될 수 있는 다양한 정보를 업로드하고, 사용자의 동작 제어를 제공하는 nCube:Thyme for Java 보드를 제작이 가능하며, nCube:Thyme for Java 를 활용한 Co2 농도의 측정 및 업로드, LED 제어 등의 기능을 본 문서를 기반으로 개발 가능하다.



nCube:Thyme for Java 에 다양한 센서를 연결하여 생활 속 곳곳에서 온도, 습도, Co2 농도, 적외선을 이용한 특정 공간에서의 사람의 존재 유무나 이동방향 등을 탐지할 수 있는 저전력 IoT Device 제작이 가능하다.

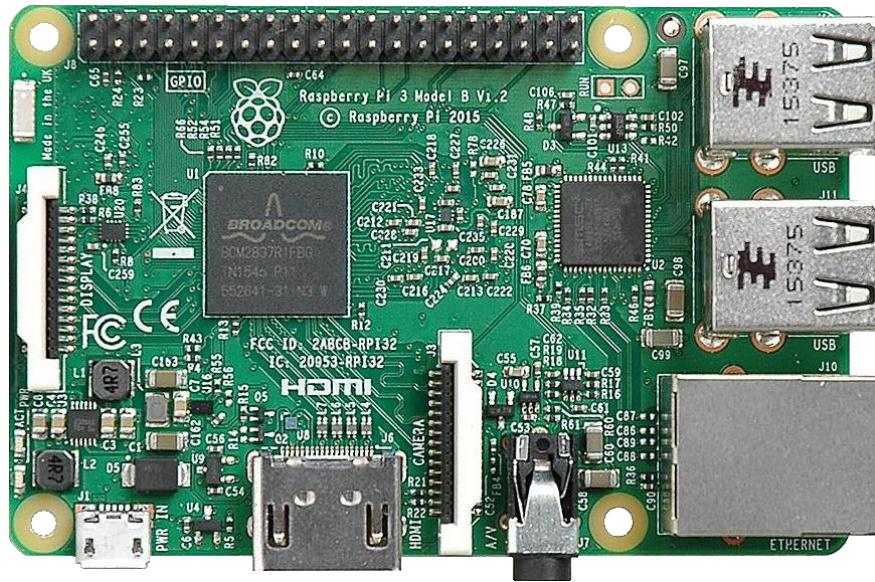
목차

1 하드웨어	4
1 - 1 Raspberry pi 3 Model B	4
2 환경설정	5
2 - 1 Raspbian OS 설치	5
2 - 2 Java 설치	9
2 - 3 Samba FTP 서버 설치	11
2 - 4 nCube:Thyme for Java 다운로드..... 오류! 책갈피가 정의되어 있지 않습니다.	
2 - 5 nCube:Thyme for Java 설정..... 오류! 책갈피가 정의되어 있지 않습니다.	
3 nCube:Thyme for Java 구동 실습	오류! 책갈피가 정의되어 있지 않습니다.
Appendix A.	20

1 하드웨어

1 장에서는 nCube:Thyme for Java 를 구성하기 위한 하드웨어에 대해 간단히 설명한다.

1 - 1 Raspberry pi 3 Model B



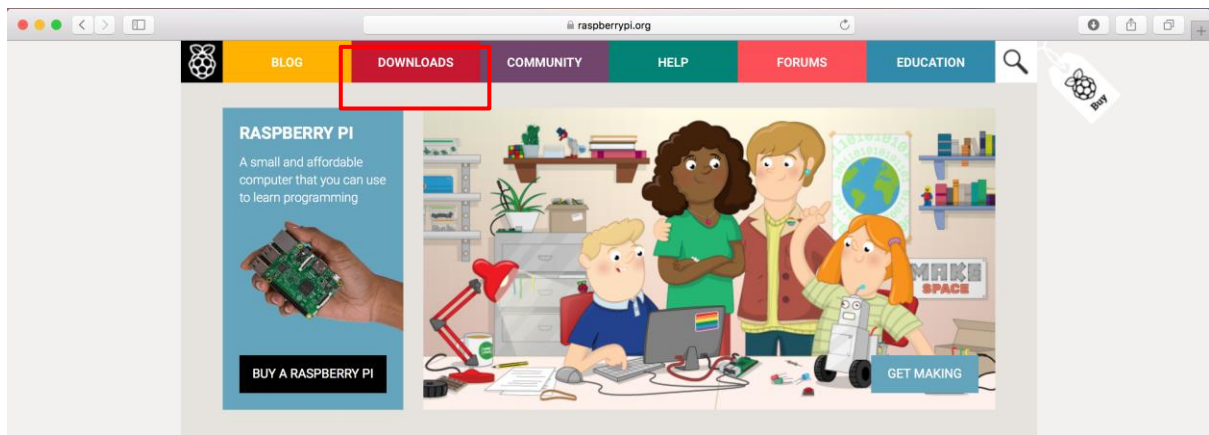
Raspberry pi 3 Model B 는 1.2GHz 로 동작하는 Broadcom 의 BCM2837 64Bit Quad Core Processor 를 탑재하고 있으며, 1GB SDRAM 과, 4 개의 USB 포트, GPIO(General Purpose Input/Output)40 pin, 이더넷 포트와 WiFi, Bluetooth LE 를 지원한다. 전원은 5pin micro USB 를 통해 5V, 2.5A 를 입력받는다.

NOOBS, Ubuntu Mate, Windows 10 IoT Core, RISC 등의 OS 를 지원하며, 본 문서에서는 Raspbian 을 설치하여 사용한다.

2 환경설정

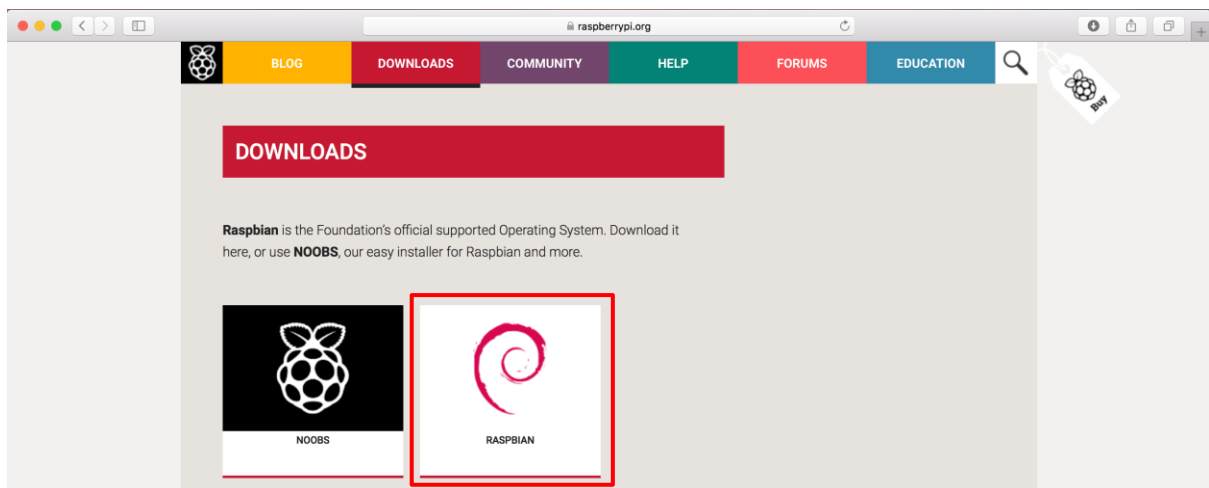
2 장에서는 nCube:Thyme for Java 보드를 제작하기 위한 환경설정에 대해 설명한다. Raspbian OS 의 설치, Java 설치, samba FTP 서버 설치, Windows 에서의 JDK, eclipse IDE 설치와 nCube:Thyme for Java 전송의 내용으로 구성하였다.

2 - 1 Raspbian OS 설치

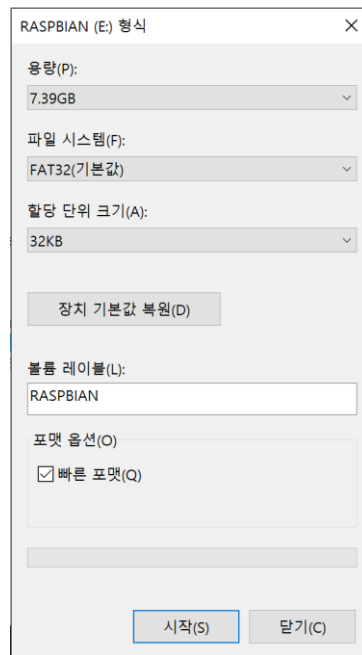


<https://www.raspberrypi.org>

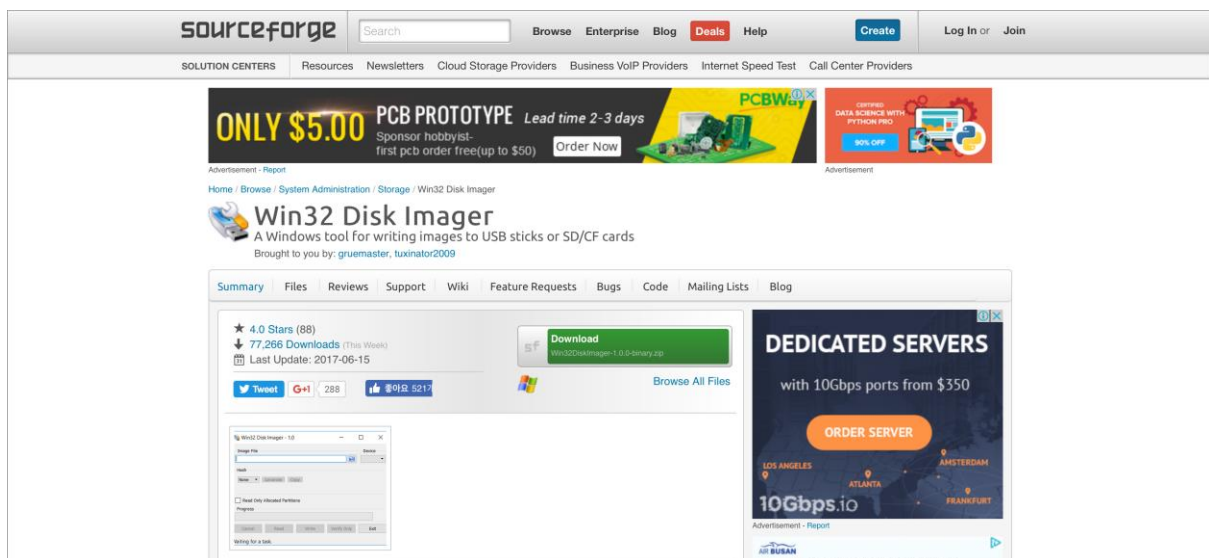
위의 URL 을 통해 Raspberry pi 홈페이지에 접속하고, DOWNLOADS 탭을 클릭한다.



RASPBIAN 아이콘을 클릭하여 Raspbian OS 를 다운로드 받는다.



8GB 이상의 microSD 를 준비하고, FAT32 형식으로 포맷한다.



<https://sourceforge.net/projects/win32diskimager/>
위의 URL 을 통해 win32 disk imager 를 다운로드 받고, 실행시킨다.

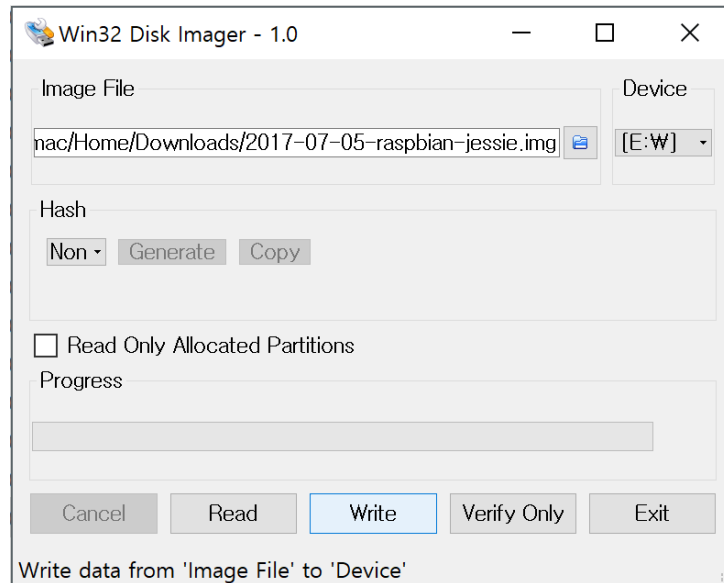


Image File 항목에 다운로드받은 Raspbian OS 이미지를 선택하고, Device 항목에 포맷한 microSD 를 선택하고, Write 버튼을 클릭하여 Raspbian OS 를 microSD 에 덮어씌운다.

작업이 완료된 후 microSD 를 Raspberry pi 3 에 삽입하고, 전원을 연결한다.

```
deory — pi@raspberrypi: ~ — ssh pi@192.168.0.98 — 80x24
[Deoryui-MacBook-Pro:~ deory$ ssh pi@192.168.0.98
[pi@192.168.0.98's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jul  5 21:06:55 2017

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~ $
```

원격으로 Raspberry Pi 3 에 접근하여 작업하기 위해 터미널의 ssh 명령어 사용 또는, putty 를 설치를 통한 ssh 를 사용한다. Host 주소는 Raspberry pi 3 의 ip 주소이며, 계정은 pi, 비밀번호는 raspberry 이다.

```
pi@raspberrypi ~ $ sudo apt-get update
.....
Reading package lists... Done
pi@raspberrypi ~ $ sudo apt-get upgrade
.....
```

Java 와 samba 설치 이전에 repository update 와 패키지 upgrade 를 하도록 한다. 각 명령어는 위의 그림을 참조한다.

2 - 2 Java 설치

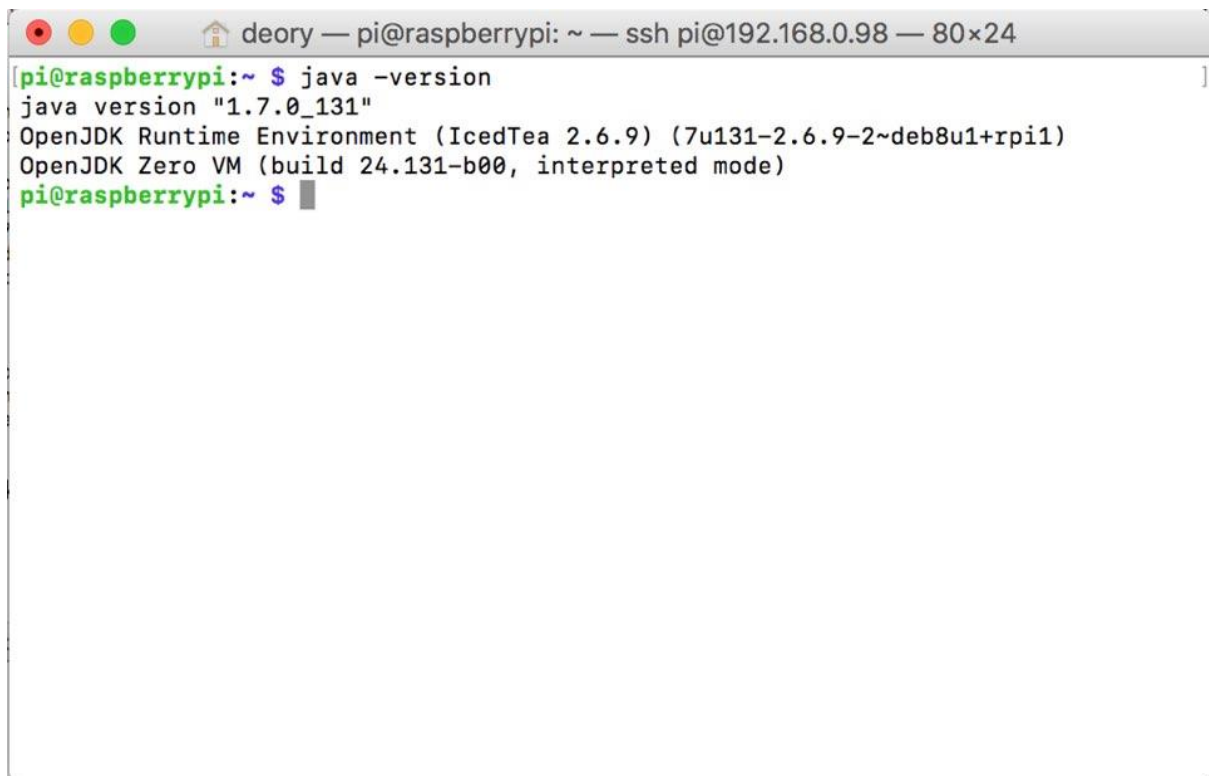
nCube:Thyme for Java 구동을 위해 Java 패키지를 설치한다.

```
pi@raspberrypi ~$ mkdir node
pi@raspberrypi ~$ cd node
pi@raspberrypi ~/node$ sudo apt-get install openjdk-7-jdk
pi@raspberrypi ~/node$ java -version (버전 확인 명령어)
```

최신 Raspbian OS 의 경우 jdk(Java Developer's Kit)가 이미 설치되어있다. 최신버전의 Java 가 필요한 경우가 아니라면 그대로 사용해도 무관하다.

```
deory — pi@raspberrypi: ~ — ssh pi@192.168.0.98 — 90x25
pi@raspberrypi:~$ sudo apt-get install openjdk-7-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libc-ares2 libv8-3.14.5
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  ca-certificates-java icedtea-7-jre-jamvm libatk-wrapper-java libatk-wrapper-java-jni
  libice-dev libpthread-stubs0-dev libsctp1 libsm-dev libx11-dev libx11-doc libxau-dev
  libxcb1-dev libxdmcp-dev libxt-dev lksctp-tools openjdk-7-jre openjdk-7-jre-headless
  tzdata-java x11proto-core-dev x11proto-input-dev x11proto-kb-dev xorg-sgml-doctools
  xtrans-dev
Suggested packages:
  libice-doc libsm-doc libxcb-doc libxt-doc openjdk-7-demo openjdk-7-source visualvm
  icedtea-7-plugin sun-java6-fonts fonts-ipafont-gothic fonts-ipafont-mincho
  ttf-wqy-microhei ttf-wqy-zenhei ttf-indic-fonts fonts-indic
```

sudo apt-get install openjdk-7-jdk 명령어를 이용하여 Java 를 설치하는 과정이다.

A terminal window titled 'deory — pi@raspberrypi: ~ — ssh pi@192.168.0.98 — 80x24'. The prompt is 'pi@raspberrypi:~ \$'. The command 'java -version' has been entered, and the output is displayed: 'java version "1.7.0_131"', 'OpenJDK Runtime Environment (IcedTea 2.6.9) (7u131-2.6.9-2~deb8u1+rpil)', and 'OpenJDK Zero VM (build 24.131-b00, interpreted mode)'. The prompt is now 'pi@raspberrypi:~ \$' with a cursor.

```
[pi@raspberrypi:~ $ java -version  
java version "1.7.0_131"  
OpenJDK Runtime Environment (IcedTea 2.6.9) (7u131-2.6.9-2~deb8u1+rpil)  
OpenJDK Zero VM (build 24.131-b00, interpreted mode)  
pi@raspberrypi:~ $
```

Java 설치가 완료되어, 버전 확인을 통해 Java 설치가 완료된 것을 확인하였다.

2 - 3 Samba FTP 서버 설치

개발환경에서 Raspberry pi 3 와 파일공유를 위해 Samba FTP 서버를 설치한다.

```
pi@raspberrypi ~ $ sudo apt-get install samba samba-common-bin
.....
Do you want to continue [Y/n]? Y
```

```
deory — pi@raspberrypi: ~ — ssh pi@192.168.0.98 — 90x24
pi@raspberrypi:~ $ sudo apt-get install samba samba-common-bin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libc-ares2 libv8-3.14.5
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  attr libaio1 libasn1-8-heimdal libfile-copy-recursive-perl libhcrypto4-heimdal
  libhdb9-heimdal libheimbase1-heimdal libhx509-5-heimdal libkrb5-26-heimdal
  libroken18-heimdal libwind0-heimdal python-crypto python-dnspython python-ldb
  python-ntdb python-samba python-tdb samba-dsdb-modules samba-vfs-modules tdb-tools
  update-inetd
Suggested packages:
  python-crypto-dbg python-crypto-doc bind9 bind9utils ctdb ldb-tools smbldap-tools
  winbind heimdal-clients
apt-get 명령어를 이용하여 Samba 설치를 완료하였다.
```

```
pi@raspberrypi ~ $ sudo smbpasswd -a pi
New SMB password: (원하는 패스워드 입력)
Retype new SMB password: (원하는 패스워드 입력)
Added user pi.
```

```
deory — pi@raspberrypi: ~ — ssh pi@192.168.0.98 — 90x24
pi@raspberrypi:~ $ sudo smbpasswd -a pi
New SMB password:
Retype new SMB password:
Added user pi.
pi@raspberrypi:~ $
```

smbpasswd 명령어를 이용하여 Samba 의 새 사용자를 등록하였다.

```
pi@raspberrypi ~ $ sudo nano /etc/samba/smb.conf
```

..... (가장 마지막 줄 밑에)

```
[pi]
```

```
comment = raspberry pi folder
```

```
path = /home/pi
```

```
valid users = pi
```

```
writable = yes
```

```
browseable = yes
```

```
<Ctrl>+<X> → Y → <Enter>
```

```
deory — pi@raspberrypi: ~ — ssh pi@192.168.0.98 — 90x24
GNU nano 2.2.6 File: /etc/samba/smb.conf

# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
; write list = root, @lpadmin

[pi]
comment = raspberry pi folder
path = /home/pi
valid user = pi
writable = yes
browseable = yes
```

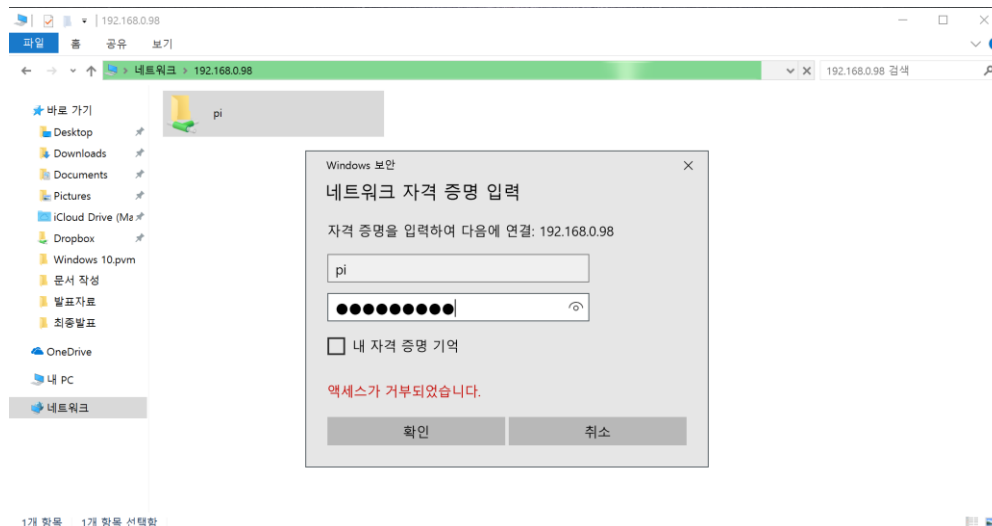
/etc/samba/smb.conf 파일을 수정하여 사용자 설정을 완료하였다.

```
pi@raspberrypi ~ $ sudo service smbd restart
```

```
deory — pi@raspberrypi: ~ — ssh pi@192.168.0.98 — 90x24
pi@raspberrypi:~ $ sudo service smbd restart
pi@raspberrypi:~ $
```

sudo service smbd restart 명령어를 통해 변경된 사용자 설정이 반영될 수 있도록 samba 를 재시작 하였다.

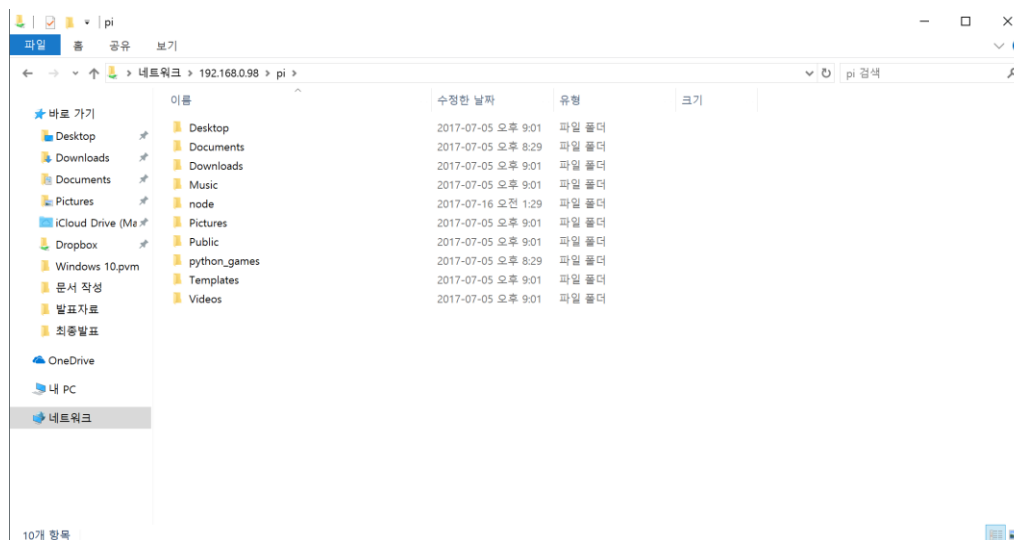
nCube:Thyme for Java 개발 가이드



Windows 탐색기의 주소입력 창에

WWRaspberry pi ip 주소 (ex. WW192.168.0.98)

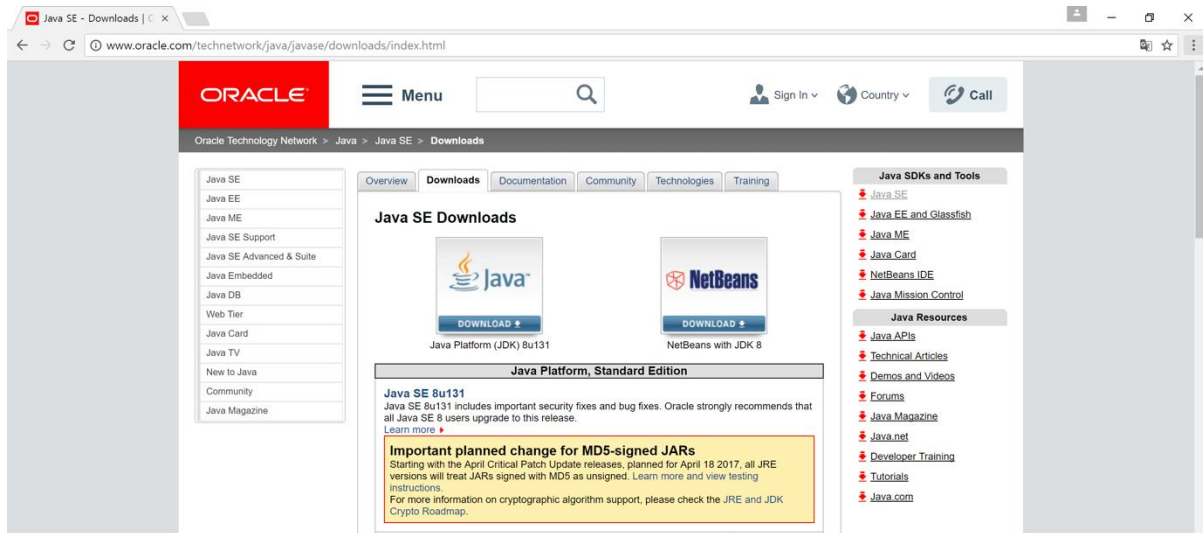
를 입력하고, 계정과 비밀번호를 각각 pi 와 samba 사용자 추가 시 입력하였던 비밀번호를 입력한다.



Raspberry pi 의 /home/pi 디렉토리에 접근이 가능한것을 확인할 수 있다.

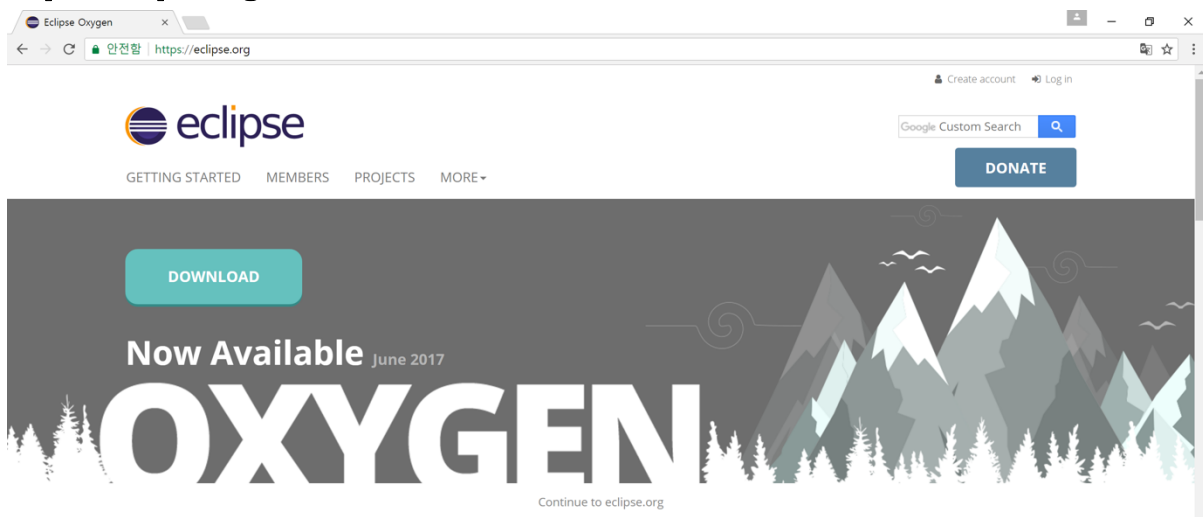
2 - 4 Windows 환경설정

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>



위의 URL 을 통해 ORACLE 사이트에서 Java SE JDK 를 다운로드 받고, 설치한다.

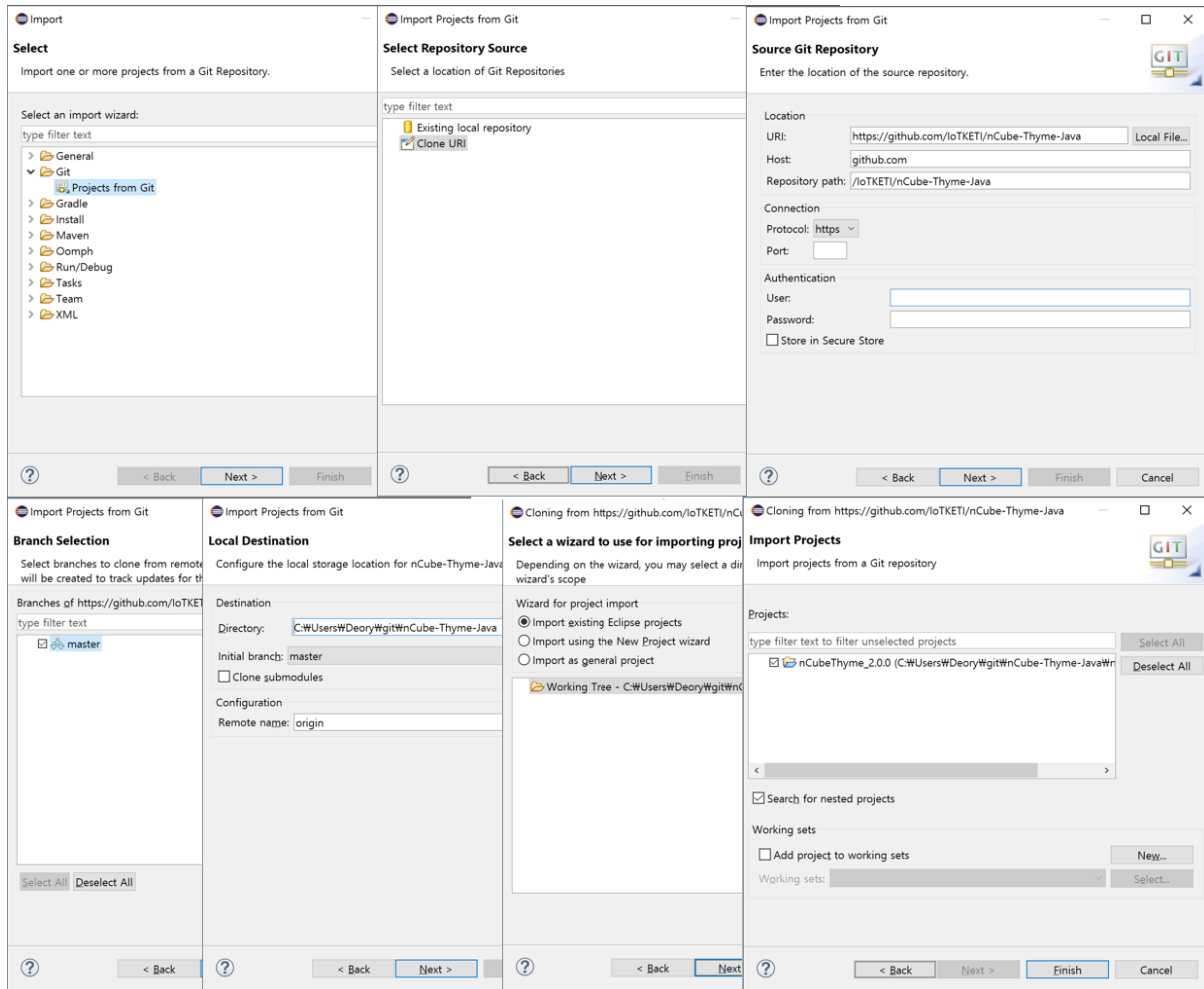
<https://eclipse.org>



위의 URL 을 통해 eclipse 사이트에서 eclipse IDE 를 다운로드 받고, 설치한다.

eclipse IDE 를 실행시키고, File 탭의 import 항목을 클릭한다.

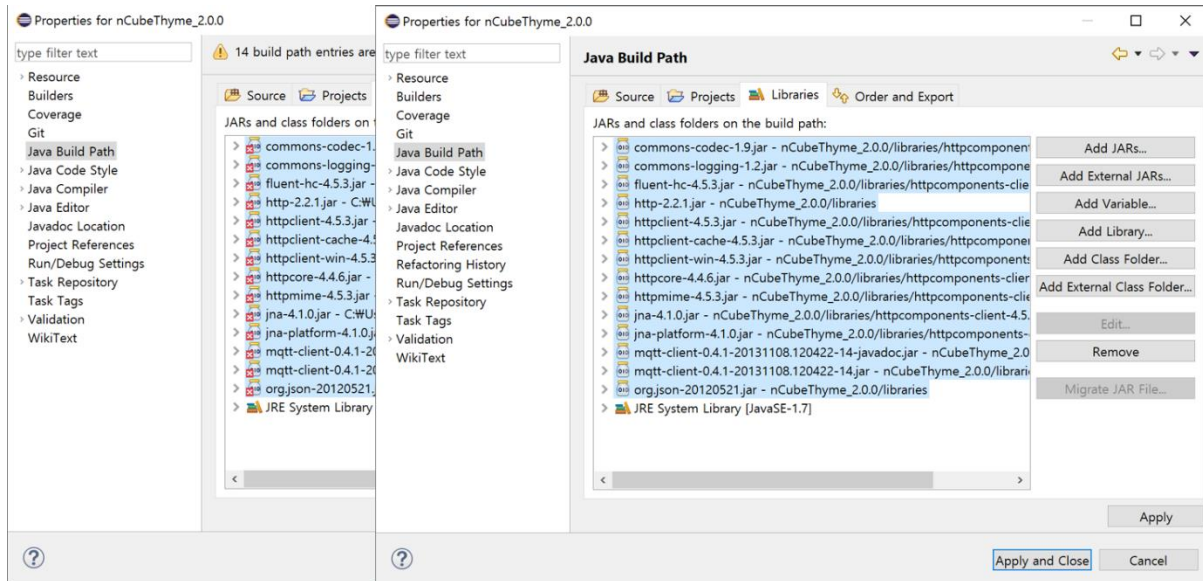
nCube:Thyme for Java 개발 가이드



Git, Project form Git, Clone URI 를 순서대로 선택하고,

<https://github.com/loTKETI/nCube-Thyme-Java>

URI 를 입력하여 진행하여 Java 프로젝트를 추가한다.

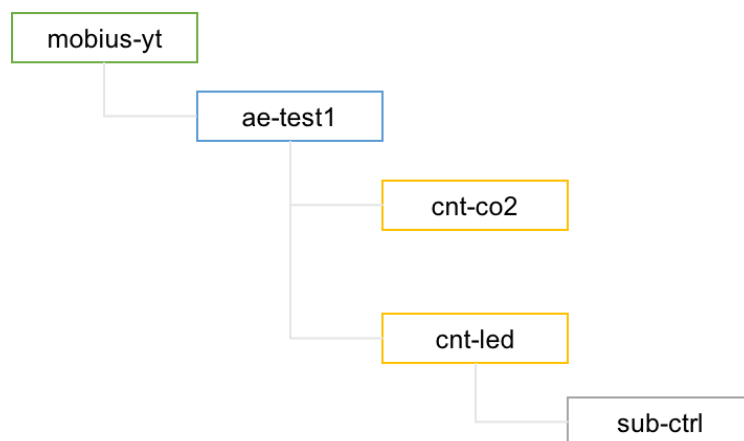


library 경로 문제를 해결하기 위해, nCube-Thyme-Java 프로젝트의 Properties 에서 Java Build Path 항목 아래의 Libraries 탭에 있는 모든 라이브러리를 제거하고, Add JARs 를 클릭하여 Library 들을 다시 추가해주어야 한다.

2 – 5 nCube:Thyme for Java Configuration

```
{
  "useprotocol": "http",
  "cse": {
    "cbhost": "203.253.128.161",
    "cbport": "7579",
    "cbname": "Mobius",
    "cbcseid": "/Mobius",
    "mqtport": "1883"
  },
  "ae": {
    "aeid": "S",
    "appid": "0.2.481.1.1",
    "appname": "ae-test1",
    "appport": "9727",
    "bodytype": "xml",
    "tasport": "3105"
  },
  "cnt": [
    {
      "parentpath": "/ae-test1",
      "ctname": "cnt-co2"
    },
    {
      "parentpath": "/ae-test1",
      "ctname": "cnt-led"
    }
  ],
  "sub": [
    {
      "parentpath": "/ae-test1/cnt-led",
      "subname": "sub-ctrl",
      "nu": "mqtt://AUTOSSET"
    }
  ]
}
```

nCube:Thyme for Java 의 conf.json 파일의 내용은 위와 같다.

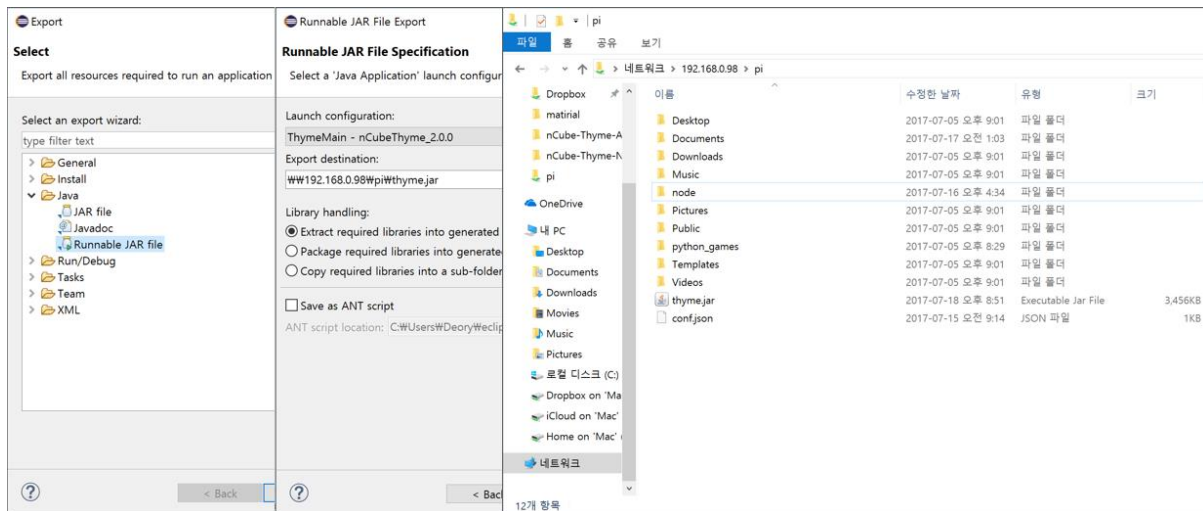


nCube:Thyme for Java 가 IoT Platform 에 생성할 리소스 구조는 위의 그림과 같이 나타낼 수 있다.

3 nCube:Thyme for Java 구동 실습

이번 장에서는 nCube:Thyme for Java 를 직접 구동시키고, Mobius Resource Monitor 를 이용하여 nCube:Thyme for Java 가 생성한 resource 를 조회해 본다.

eclipse IDE 를 이용하여 import 한 nCube:Thyme for Java 의 ThymeMain Application 을 1 회 이상 실행시킨다.



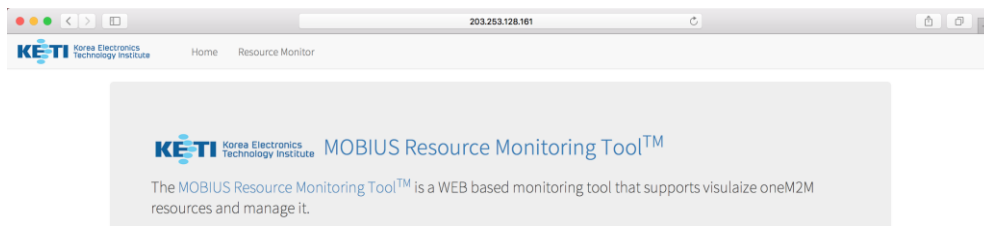
eclipse IDE 의 File 탭을 누르고 Export 항목을 클릭한다. Export 창에서 Java, Runnable JAR file 을 선택한다. Launch configuration 을 클릭하여 ThymeMain 항목을 선택하고, Export destination 에 Raspberry Pi 의 Pi 홈 디렉토리 경로를 적어준다. Export 가 완료된 후에 conf.json 파일도 같은 위치로 이동시켜준다.

```
deory — pi@raspberrypi: ~ — ssh pi@192.168.0.98 — 90x25
pi@raspberrypi:~ $ ls
conf.json  Documents  Music  Pictures  python_games  thyme.jar
Desktop    Downloads  node   Public   Templates     Videos
pi@raspberrypi:~ $ java -jar thyme.jar
[&CubeThyme] &CubeThyme SW start.....

[&CubeThyme] Configuration file loading...
[&CubeThyme] CSE - cbhost : 203.253.128.161
[&CubeThyme] CSE - cbport : 7579
[&CubeThyme] CSE - cbname : Mobius
[&CubeThyme] CSE - cbcseid : /Mobius
[&CubeThyme] CSE - mqttPort : 1883
```

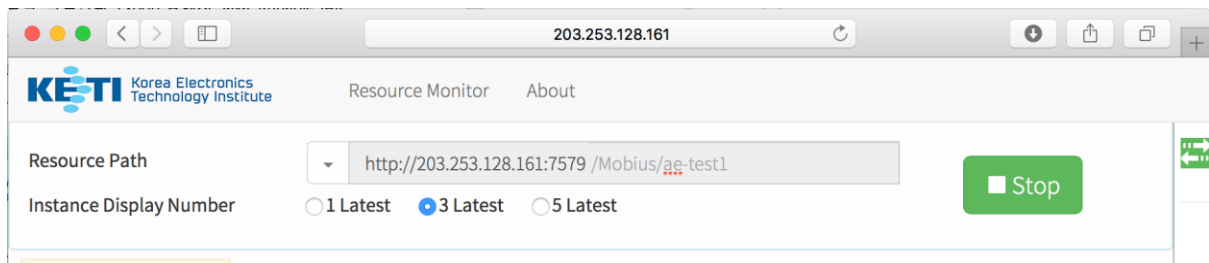
java -jar thyme.jar 명령어를 이용하여 nCube:Thyme for Java 를 실행시킨다.

<http://203.253.128.161:7575/>

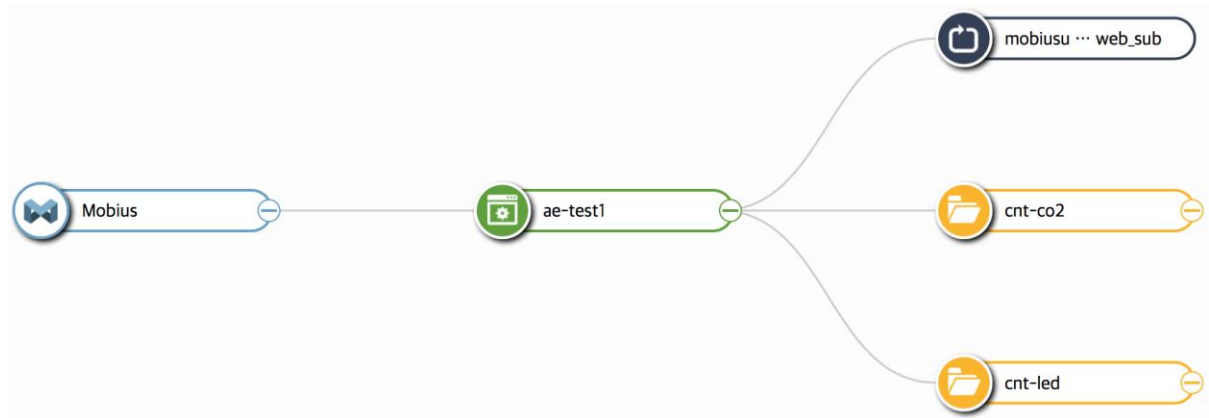


위의 URL 을 통해 웹버전의 Mobius Resource Monitor 에 접속한다.
Resource Monitor 기능을 이용하기 위해 상단의 메뉴에서 Resource Monitor 탭을
클릭하여 Resource Monitor 페이지에 접속한다.

<http://203.253.128.161:7579/Mobius/ae-test1>



Resource Path 에 위의 주소를 넣고 Start 버튼을 클릭하면, nCube:Thyme for Java 가
생성한 Resource Tree 를 볼 수 있다.



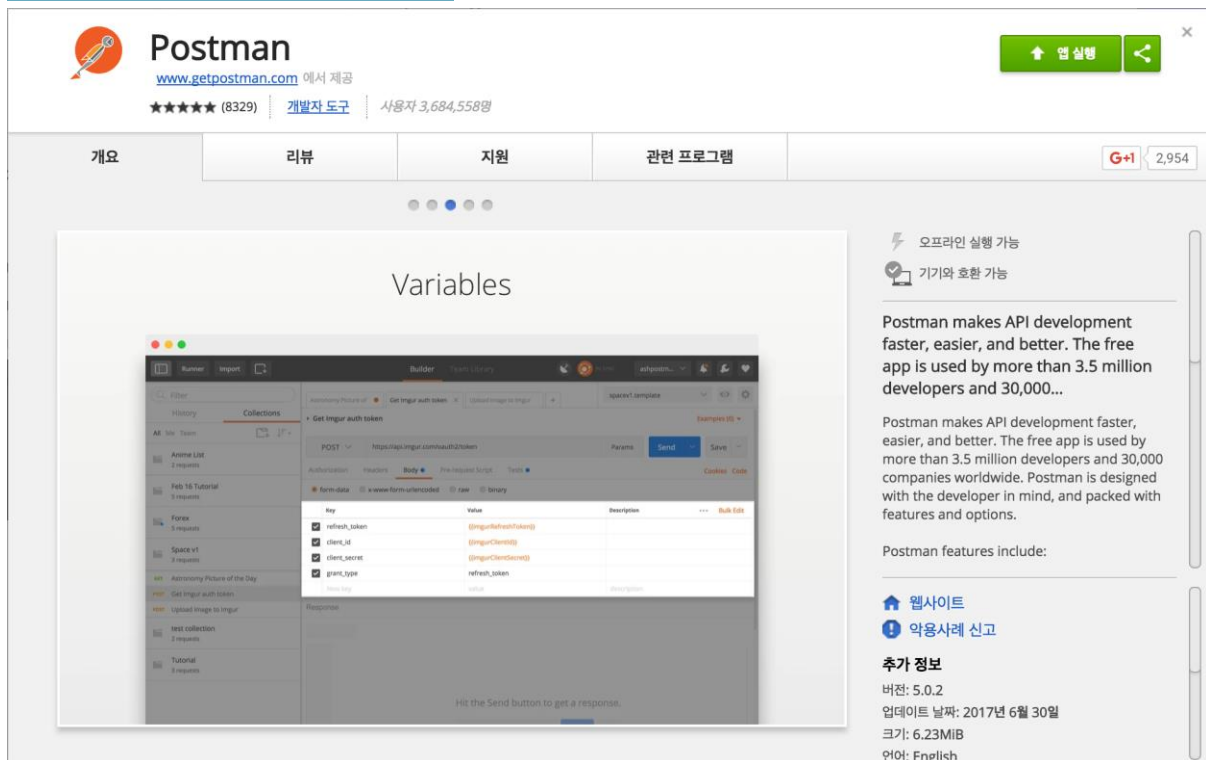
조회된 리소스 구조는 위의 그림과 같다.

Appendix A.

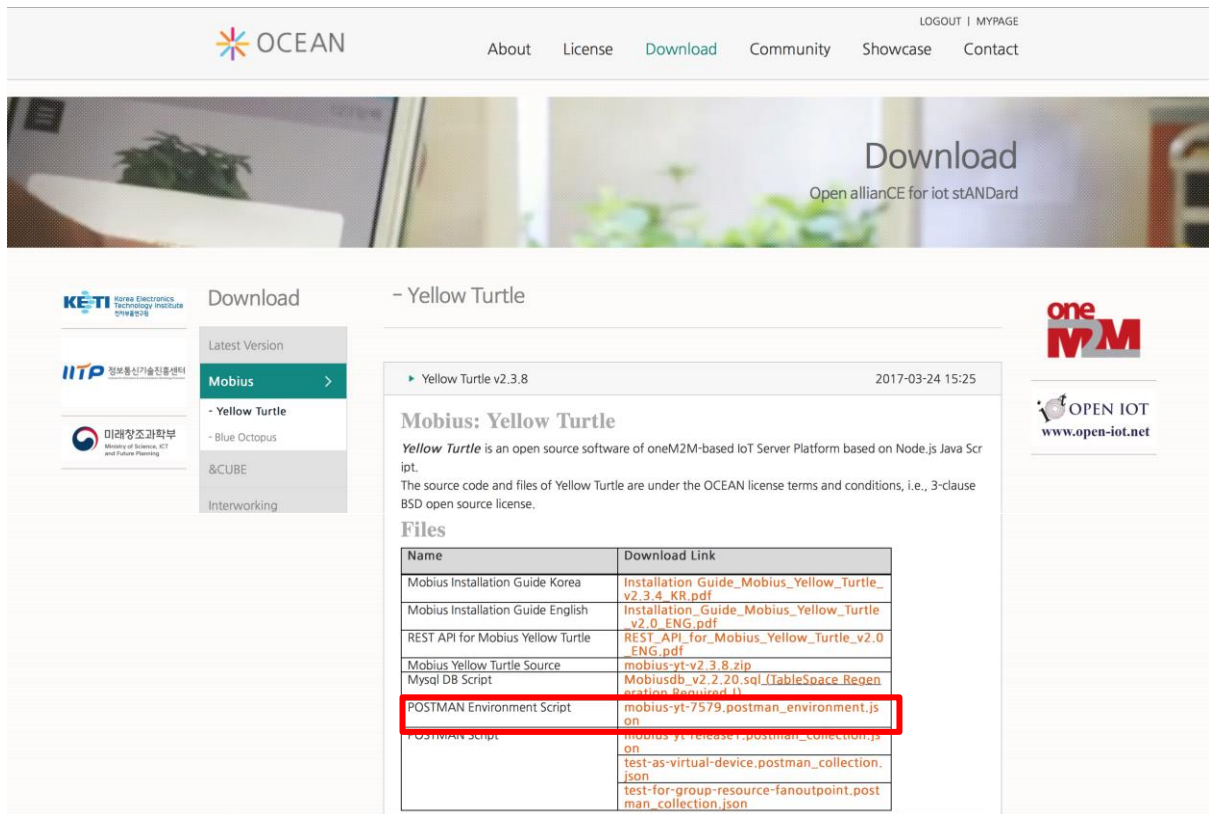
Postman 을 활용한 nCube:Thyme for Java Resource 조회와 제어

Google Chrome 의 Postman 앱을 이용해서 nCube:Thyme for Java 가 생성한 resource 조회가 가능하다. 이외에도 API 를 활용한 AE, CNT, CIN 생성 및 조회가 가능하다.

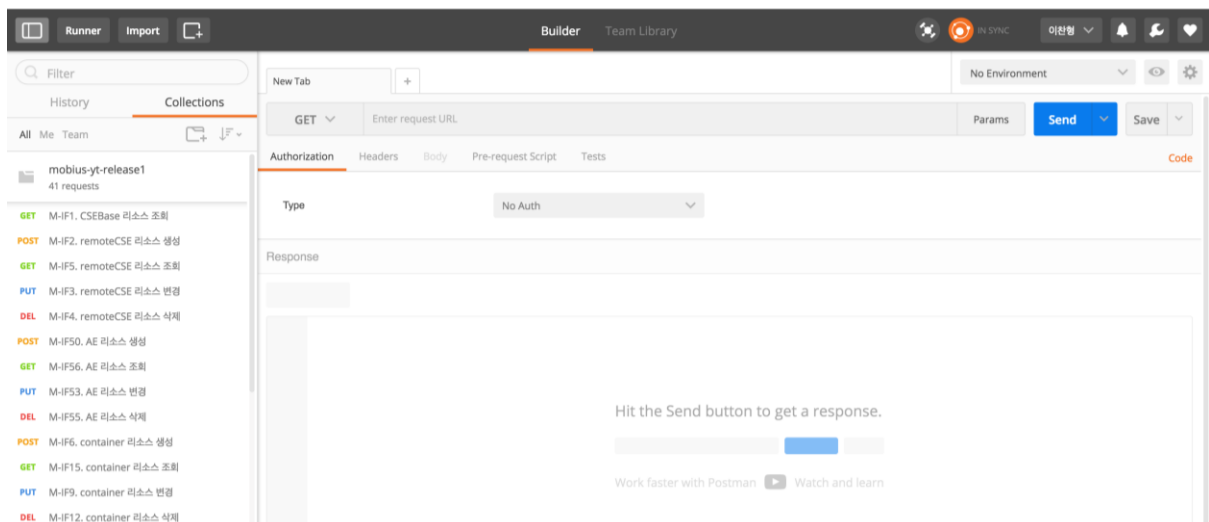
https://chrome.google.com/webstore/detail/postman/fhbjgbfijnjbdgggehcdcbncdddomo?utm_source=chrome-ntp-icon



Google Chrome 을 이용해서 위의 URL 에 접속해 Postman 앱을 다운로드 및 설치하고, 실행시킨다.



http://www.iotocean.org 의 Download, Mobius Yellow Turtle, POSTMAN Environment Script 아래의 mobius-yt-7579.postman_environment.json 을 클릭하여 Postman API collection 을 다운로드 받는다.



Postman에서 import → import file → choose files 를 클릭하여 위에서 다운로드 받은 API collection 을 열면 Postman 좌측의 Collections 목록 아래에 API들이 추가된 것을 볼 수 있다. 추가된 API들을 이용하여 resource 조회, 생성이 가능하다.