

Mobius 오픈 IoT 플랫폼

환경, 준비사항

서비스 다운로드 경로

서비스	Git 주소
mobius_open_iot	https://github.com/loTKETI/mobius-open-platform

서비스 별 기본 포트

서비스 구성	Port
Webportal	8881
Dashboard	7580
Resource Browser	7575
SNS agent manage tool	7597
OTA manage tool	8730

운영체제 및 프로그램 버전

OS 및 기타 프로그램	버전
OS	Linux Ubuntu 18.04 LTS
NodeJS	12.16.1
Npm	6.13.4
MongoDB	4.2.3
PM2	4.2.3
Gulp-cli	2.2.0

서비스 서버 도메인

도메인 주소
ex) iotocean.org

※주의 : 도메인 설정을 하지 않을 경우 서비스에 로그인할 수 없습니다.(localhost 사용불가)

Node.js 설치



오픈 IoT 플랫폼은 Javascript 기반으로 이루어진 Web기반 서비스이며 서비스를 실행시키기 위해 Node.js 필요함에 따라 설치를 진행한다. 설치방법은 아래와 같다.

Node.js 설치

```
$ curl -sL https://deb.nodesource.com/setup_10.x | sudo bash -  
$ sudo apt install Node.js
```

Node.js 업데이트

```
$ sudo npm cache clean -f  
$ sudo npm install -g n  
$ sudo n stable
```

Node.js 및 NPM 버전 확인

```
$ sudo node -v  
$ sudo npm -v
```

A screenshot of a terminal window with a black background and green text. The prompt is 'kjh@KJH-G531: ~'. The first command is 'sudo node -v' and the output is 'v12.16.1'. The second command is 'sudo npm -v' and the output is '6.13.4'.

```
kjh@KJH-G531: ~  
kjh@KJH-G531:~$ sudo node -v  
v12.16.1  
kjh@KJH-G531:~$ sudo npm -v  
6.13.4
```

<그림-1>Node.js 및 npm 설치 확인



MongoDB는 IoT 플랫폼의 데이터 저장을 위해 사용되는 데이터베이스이다. 설치방법은 MongoDB의 공식 매뉴얼을 따른다.

패키지의 일관성과 훼손을 막기 위해 배포자가 서명한 GPG 키 불러오기

```
$ wget -qO - https://www.mongodb.org/static/pgp/server-4.2.asc | sudo apt-key add -
```

```
kjh@KJH-G531:~$ sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 9DA31620334BD75D9DCB49F368818C72E52529D4
[sudo] password for kjh:
Executing: /tmp/apt-key-gpghome.DcPsLrt0iR/gpg.1.sh --keyserver hkp://keyserver.ubuntu.com:80 --recv 9DA31620334BD75D9DCB49F368818C72E52529D4
gpg: key 68818C72E52529D4: public key "MongoDB 4.0 Release Signing Key <packaging@mongodb.com>" imported
gpg: Total number processed: 1
gpg: imported: 1
```

<그림-2> GPG 키를 성공적으로 가져온 화면

MongoDB 패키지 저장소 등록

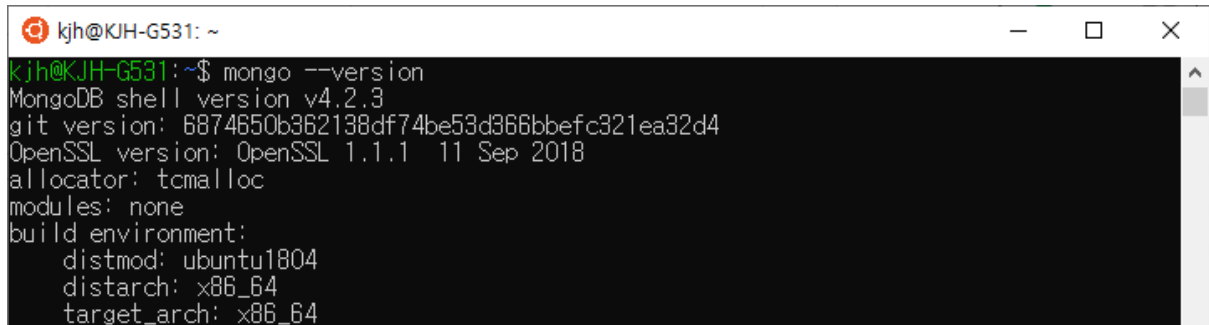
```
$ echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.2 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.2.list
```

패키지 업데이트 후 MongoDB 설치

```
$ sudo apt-get update
$ sudo apt-get install -y mongodb-org
```

MongoDB 설치 확인

```
$ mongo --version
```

A terminal window titled 'kjh@KJH-G531: ~' showing the output of the 'mongo --version' command. The output lists various details about the MongoDB shell version v4.2.3, including git version, OpenSSL version, allocator, modules, and build environment details like distmod, distarch, and target_arch.

```
kjh@KJH-G531:~$ mongo --version
MongoDB shell version v4.2.3
git version: 6874650b362138df74be53d366bbefc321ea32d4
OpenSSL version: OpenSSL 1.1.1 11 Sep 2018
allocator: tcmalloc
modules: none
build environment:
  distmod: ubuntu1804
  distarch: x86_64
  target_arch: x86_64
```

<그림-3> MongoDB 설치 확인

MongoDB 관리자 계정생성

```
mongo
use admin
db.createUser({
  user : <관리자 계정명>,
  pwd : <계정 비밀번호>,
  roles : [
    {role : "userAdminAnyDatabase", db : "admin"}
  ]
})
```

MongoDB 보안 인증 켜기

```
$ sudo vi /etc/mongod.conf

...
31 security :
    authorization: enable
...
```

#security에서 #을 지워 주석을 제거 후 authorization : enable을 입력한다.

MongoDB 서비스 재시작

```
$ sudo service mongod restart
```

Gulp & PM2 설치

Gulp는 Node.js 기반의 Task Runner이며, 본 플랫폼 설치를 간단하게 설치하기 위해 사용되는 라이브러리이다.



PM2는 Node.js 프로세스 관리 도구이다. Gulp가 주로 설치나 설정을 위한 용도로 사용된다면, PM2는 Node.js 애플리케이션을 시작/중지/재시작 등 관리하는 도구이다.



설치방법은 아래와 같다. NPM에서 --global 옵션으로 설치된 라이브러리들은 '전역'설치라 부르며 global옵션으로 설치된 라이브러리는 명령어처럼 사용이 가능하다.

Gulp-cli, PM2 설치

```
$ sudo npm install --global gulp-cli pm2
```


APT 저장소 추가

```
$ sudo su
$ sudo vi /etc/apt/sources.list
...
# Nginx 추가
deb http://nginx.org/packages/mainline/ubuntu/ xenial nginx
deb-src http://nginx.org/packages/mainline/ubuntu/ xenial nginx
```

APT 저장소 보안키 다운로드 / 등록

```
$ cd /root
$ wget http://nginx.org/keys/nginx_signing.key
$ apt-key add nginx_signing.key
$ rm nginx_signing.key
```

APT 소스 패키지 정보 업데이트

```
$ apt-get update
```

Nginx 설치

```
$ apt-get install nginx
```

Nginx 실행

```
$ service nginx restart
$ nginx -v
```

Nginx 설정

서비스	설명
webportal	location ^~ /webportal { proxy_pass http://localhost:8881 }
디바이스 관리	Mobius 와 연결되어 있는 디바이스를 추가 및 관리하는 서비스

서비스 구동

소개

서비스 주소 : https://github.com/loTKETI/mobius_open_platform



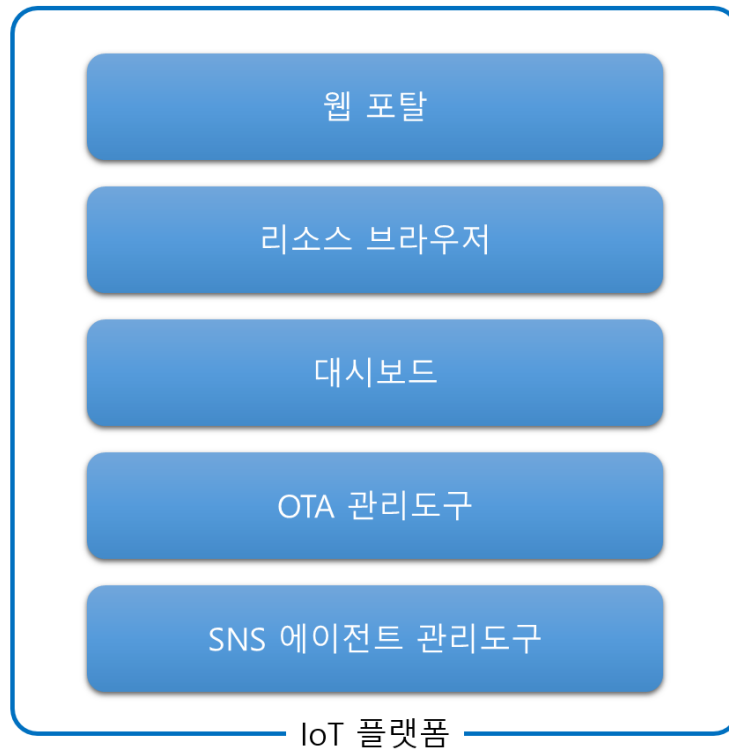
<그림-6> IoT 플랫폼 제공 서비스

IoT 플랫폼에는 8 가지 서비스가 제공되고 있다. 각 서비스에 대한 정의는 아래와 같다.

서비스	설명
Mobius 상태 조회	현재 Mobius 상태를 시각화 하여 보여주는 서비스
디바이스 관리	Mobius 와 연결되어 있는 디바이스를 추가 및 관리하는 서비스
접근 권한 관리	사용자가 등록한 Mobius 자원에 대한 접근을 제어하는 서비스
데이터 탐색기	사용자가 등록한 디바이스의 데이터를 조회하는 서비스
리소스 브라우저	Mobius 자원 구조를 시각화 하여 트리 그래픽으로 보여주는 기능
대시보드	사용자가 생성한 조건에 맞추어 Mobius 의 자원 상태를 Widget 으로 보여주는 서비스
OTA 관리기	Mobius 와 연결된 nCube 을 대상으로 원격 펌웨어 업데이트 제공
에이전트 관리기	Mobius 와 상호작용이 가능한 Telegram Bot 을 등록, 명령어 생성 및 관리하는 서비스

시스템 구동

IoT 플랫폼은 총 5 개의 모듈로 구성되며, 각 모듈은 웹 포탈, 리소스 브라우저, 대시보드, OTA 관리, 에이전트 관리로 이루어져 있다.



<그림-7> 플랫폼 모듈 구성

서비스	설명
웹 포탈	Mobius 상태조회, 디바이스 관리, 접근 권한관리, 데이터 탐색 등 Mobius 에 대한 정보 확인 및 설정 등 관리가 가능하다.
리소스 브라우저	Mobius 내에 있는 AE 를 조회하여 내부 구조를 보기 쉽게 보여주며 실시간 업데이트를 제공한다.
대시보드	사용자가 생성한 알고리즘의 결과물을 Widget 으로 보여주는 기능을 제공한다.
OTA 관리도구	OTA(Over The Air) 시스템으로 Mobius 와 연결된 nCube 기기를 대상으로 원격 펌웨어 업데이트를 제공한다.
SNS 에이전트 관리도구	Telegram 으로 생성된 봇을 등록하여 Mobius 와 간단한 생성/읽기/구독과 같은 상호작용을 지원한다.

설치

본 플랫폼 설치를 위해서 GitHub 에서 가져온다.

GitHub 에서 다운로드

```
$ git clone https://github.com/IoTKETI/mobius-open-platform
```

Clone 후 설치를 위해 사용되는 라이브러리를 설치

Platform NPM 라이브러리 설치

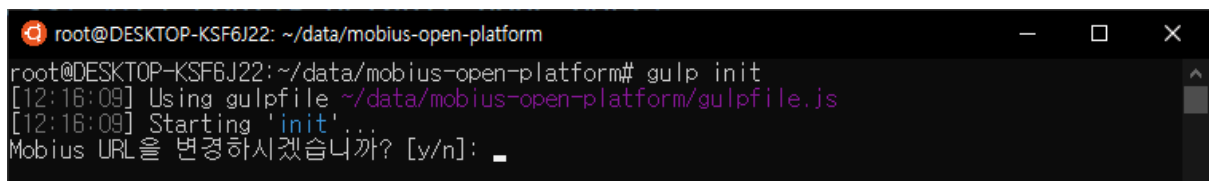
```
$ cd mobius-open-platform  
$ npm install
```

본 플랫폼은 간단한 설치를 위해 Gulp 를 사용한 명령어를 제공한다. 4 개의 설정을 마친 뒤 Database 설정과 각 서비스별 npm 설치를 자동으로 수행한다.

Platform 설정 시작

```
$ gulp init
```

1) Mobius 주소 변경

A terminal window with a dark background. The title bar shows 'root@DESKTOP-KSF6J22: ~/data/mobius-open-platform'. The command prompt shows 'root@DESKTOP-KSF6J22:~/data/mobius-open-platform# gulp init'. The output shows '[12:16:09] Using gulpfile ~/data/mobius-open-platform/gulpfile.js' and '[12:16:09] Starting 'init'...'. The prompt then asks 'Mobius URL을 변경하시겠습니까? [y/n]:' with a cursor pointing to the input field.

```
root@DESKTOP-KSF6J22: ~/data/mobius-open-platform  
root@DESKTOP-KSF6J22:~/data/mobius-open-platform# gulp init  
[12:16:09] Using gulpfile ~/data/mobius-open-platform/gulpfile.js  
[12:16:09] Starting 'init'...  
Mobius URL을 변경하시겠습니까? [y/n]:
```

첫번째 기능으로는 Mobius 의 주소를 변경하는 기능이다. Mobius URL 은 기본 값으로 KETI 의 Mobius 로 설정되어 있으나, 사용자가 설치하거나 다른 곳의 Mobius 를 참조하기 원하는 경우 위의 기능에서 설정이 가능하다.

```
root@DESKTOP-KSF6J22: ~/data/mobius-open-platform
root@DESKTOP-KSF6J22:~/data/mobius-open-platform# gulp init
[12:47:03] Using gulpfile ~/data/mobius-open-platform/gulpfile.js
[12:47:03] Starting 'init'...
Mobius URL을 변경하시겠습니까? [y/n]: y
변경할 Mobius의 주소를 입력해주세요(예 : www.mobius.com or 192.168.0.1) : localocean.org
Mobius의 포트 번호를 입력해주세요 : 8080_
```

2) 서비스 포트 변경

```
root@DESKTOP-KSF6J22: ~/data/mobius-open-platform
[1] WEBPORTAL
[2] DASHBOARD
[3] OTA
[4] SNS
[5] RES
[6] done
[0] CANCEL

포트번호를 변경할 서비스를 선택해주세요. [1...6 / 0]: _
```

두번째 기능은 각 서비스에 기본으로 할당된 포트 번호를 변경하는 기능이다. 1~5 을 입력하여 포트를 변경할 수 있으며, 6 을 입력하는 경우 다음 기능으로 이동한다, 0 을 누르는 경우에는 설치를 취소한다.

3) 서비스 주소 설정

```
root@DESKTOP-KSF6J22: ~/data/mobius-open-platform
서비스 주소를 설정합니다. 서브 도메인을 사용하는 경우 Nginx 설정과 동일하게 입력해주세요.
(ex iotocean.org : _
```

세번째 기능은 본 플랫폼에 사용할 도메인 또는 IP 를 입력하는 기능이다. 위 설정에 입력되는 주소로 모든 서비스가 연결된다.

4) 서브도메인 설정

```
root@DESKTOP-KSF6J22: ~/data/mobius-open-platform
서브도메인을 설정하시겠습니까? [y/n]: y
[1] Webportal => portal
[2] Dashboard => dashboard
[3] OTA manage tool => ota
[4] SNS agent manage tool => sns
[5] Resource Browser => res
[6] done
[0] CANCEL
서브도메인을 변경할 서비스를 선택해주세요 [1...6 / 0]: _
```

Nginx 을 설치하여 운용하는 경우 필요한 기능이며, 각 서비스별 서브도메인을 설정할 수 있다. 1~5 을 입력하여 대상 서비스를 선택하여 서브도메인을 변할 수 있으며 6 을 입력하여 설정을 마칠 수 있다. 0 을 입력하는 경우 서비스 설치를 취소한다.

5) MongoDB Admin 계정 정보 입력

```
root@DESKTOP-KSF6J22: ~/data/mobius-open-platform
MongoDB에 설정된 Admin 계정을 입력해주세요 : mobius_admin
mobius_admin의 패스워드 :
```

MongoDB 셋팅을 위해 MongoDB 설치 후 생성한 admin 계정의 정보를 입력한다. 본과정을 거치지 않으면 MongoDB 설정이 완료되지 않아 정상적으로 서비스가 동작하지 않는다.

실행

본 플랫폼은 PM2를 사용하여 각 Node.js 프로세스들을 관리한다. Gulp에 설정한 명령어를 실행하여 전 서비스를 실행한다.

서비스 PM2 실행

```
$ gulp serviceStart
```

```
[PM2] Starting /home/kjh/data/mobius-open-platform/resource_browser/bin/www in fork_mode (1 instance)
[PM2] Done.

```

id	name	namespace	version	mode	pid	uptime	⌘	status	cpu	mem	user	watching
1	dashboard	default	0.0.1	fork	302	1s	0	online	0%	73.5mb	kjh	disabled
2	ota	default	1.1.0	fork	328	0s	0	online	0%	58.7mb	kjh	disabled
4	res	default	0.0.0	fork	380	0s	0	online	0%	5.7mb	kjh	disabled
3	sns	default	1.2.2	fork	354	0s	0	online	0%	39.8mb	kjh	disabled
0	webportal	default	0.0.1	fork	276	1s	0	online	0%	67.5mb	kjh	disabled

```
[11:07:34] Finished 'startRes' after 379 ms
[11:07:34] Starting 'save'...
$ pm2 save
[PM2] Saving current process list...
[PM2] Successfully saved in /home/kjh/.pm2/dump.pm2
[11:07:34] Finished 'save' after 295 ms
[11:07:34] Finished 'serviceStart' after 2.08 s
kjh@KJH-G531: ~/data/mobius-open-platform$
```

<그림-> pm2 실행 명령어 결과

PM2 실행 명령어를 실행 후 동작 확인을 위해 PM2 목록을 조회하여 서비스 상태를 확인한다.
출력되는 테이블에서 Status를 확인하면 서비스가 정상 실행 중인지 확인이 가능하다.

PM2 실행 목록 조회

```
$ pm2 list
```

```
kjh@KJH-G531: ~/data/mobius-open-platform
kjh@KJH-G531:~/data/mobius-open-platform$ pm2 list

```

id	name	namespace	version	mode	pid	uptime	⌘	status	cpu	mem	user	watching
1	dashboard	default	0.0.1	fork	1483	23s	16	online	0%	53.8mb	kjh	disabled
2	ota	default	1.1.0	fork	1490	23s	16	online	0%	43.9mb	kjh	disabled
4	res	default	0.0.0	fork	1534	3s	0	online	15.8%	48.9mb	kjh	disabled
3	sns	default	1.2.2	fork	1497	23s	16	online	0%	48.3mb	kjh	disabled
0	webportal	default	0.0.1	fork	1476	23s	16	online	0%	49.1mb	kjh	disabled

```
kjh@KJH-G531:~/data/mobius-open-platform$
```

<그림-> 정상 실행화면

```
kjh@KJH-G531: ~/data/mobius-open-platform
kjh@KJH-G531:~/data/mobius-open-platform$ pm2 list

```

id	name	namespace	version	mode	pid	uptime	⌘	status	cpu	mem	user	watching
1	dashboard	default	0.0.1	fork	0	0	16	errored	0%	0b	kjh	disabled
2	ota	default	1.1.0	fork	0	0	16	errored	0%	0b	kjh	disabled
4	res	default	0.0.0	fork	380	4m	0	online	0%	35.7mb	kjh	disabled
3	sns	default	1.2.2	fork	0	0	16	errored	0%	0b	kjh	disabled
0	webportal	default	0.0.1	fork	0	0	16	errored	0%	0b	kjh	disabled

```
kjh@KJH-G531:~/data/mobius-open-platform$
```

<그림-> 비정상 실행화면

대부분의 경우는 Node.js 나 NPM 라이브러리 설치 및 버전문제의 경우가 대부분이기 때문에.
PM2 logs 로 확인을 한 뒤, 에러를 고친 뒤 다시 서비스를 시작하면 된다.

서비스 재실행

```
$ gulp serviceRestart
```

```
kjh@KJH-G531: ~/data/mobius-open-platform
$ pm2 restart webportal dashboard ota sns
Use --update-env to update environment variables
[PM2] Applying action restartProcessId on app [webportal](ids: [ 0 ])
[PM2] [webportal](0) ✓
Use --update-env to update environment variables
[PM2] Applying action restartProcessId on app [dashboard](ids: [ 1 ])
[PM2] [dashboard](1) ✓
Use --update-env to update environment variables
[PM2] Applying action restartProcessId on app [ota](ids: [ 2 ])
[PM2] [ota](2) ✓
Use --update-env to update environment variables
[PM2] Applying action restartProcessId on app [sns](ids: [ 3 ])
[PM2] [sns](3) ✓
```

id	name	namespace	version	mode	pid	uptime	⚙	status	cpu	mem	user	watching
1	dashboard	default	0.0.1	fork	2065	0s	16	online	0%	19.0mb	kjh	disabled
2	ota	default	1.1.0	fork	2072	0s	16	online	0%	14.4mb	kjh	disabled
4	res	default	0.0.0	fork	0	0	15	errored	0%	0b	kjh	disabled
3	sns	default	1.2.2	fork	2079	0s	16	online	0%	6.3mb	kjh	disabled
0	webportal	default	0.0.1	fork	2058	0s	16	online	0%	19.7mb	kjh	disabled

```
kjh@KJH-G531: ~/data/mobius-open-platform$
```

<그림-> 서비스 재시작 화면

이후 설정한 Domain 으로 이동하여 서비스가 정상적으로 동작하는지 확인한다.

ex) <http://portal.iotocean.org>

