



# 포팅메뉴얼 - 240627 수정

## 1.개요

### 1-1. 개발 환경

#### Backend

- **Java** : Oracle Open JDK 17.0.9
- **Spring Boot** : 3.2.3
- **JPA** : hibernate-core-6.4.4
- **DB**: MySQL 8.0.36
- **IntelliJ** : 2024. 3

#### Frontend

- **Node.js** : 20.10.0
- **TypeScript** : 4.9.5
- **React** : 18.2.0
- **Axios** : 1.6.8
- **Vscode** : 1.85.1

#### Server

- **AWS EC2**
  - **CPU** : 4CPU
  - **RAM** : 15GB
  - **OS** : Ubuntu 20.04.6 LTS

#### Service

- **NginX** : 1.18.0

- HTTPS

## 1-2. 프로젝트 사용도구

- 이슈 관리 : JIRA
- 형상 관리 : Gitlab
- 커뮤니케이션 : Notion, Mattermost
- 디자인 : Figma
- CI/CD : Jenkins

## 1-3. Gitignore 정보

- React : .env (최상단 위치)
- Spring : application.yml

# 2. 빌드

## 2-1. 환경변수

### Spring

- application.yml

```
server:
  port : 8081
spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://localhost:3306/dokebi?useSSL=false&allow
    username: dokebi
    password: [직접 설정]
  jpa:
    hibernate:
      ddl-auto: update
    hikari:
      idle-timeout: 300
      max-lifetime: 300
```

```

properties:
  hibernate:
    format_sql: true

  show_sql: true
  defer-datasource-initialization: true
sql:
  init:
    mode: always
output:
  ansi.enabled: always

kakao:
  key: [kakao api key]
  request-url: https://kauth.kakao.com/oauth/token
  user-info-url: https://kapi.kakao.com/v2/user/me
  redirect-url: http://localhost:8080/api/member/login/kakao

jwt:
  secret: [secret token]

email:
  address: kewer9024@ajou.ac.kr
  password: onfq ycsv evtc bjvo

aes256:
  secret: [secret key]
  iv: [iv]

```

## React

- .env : 최상단 위치

```

REACT_APP_HOME_URL=http://localhost:8080 ( or 도메인 주소 )
REACT_APP_KAKAOMAP_KEY=
REACT_APP_KAKAO_OAUTH_KEY=

```

## Python

- requirements.txt : src 내 위치

```
Flask==2.2.5
Flask-SQLAlchemy==3.0.2
mysql-connector-python==8.3.0
numpy==1.26.4
openpyxl==3.0.10
pandas==2.1.4
scikit-learn==1.4.1.post1
scipy==1.12.0
SQLAlchemy==2.0.2
5PyMySQL==1.1.0
```

## 3. 배포하기

Linux기준

### git 설치

```
sudo apt-get install git

sudo apt install git

# 버전확인
git --version
```

clone받기

**https 접근을 위한 환경을 세팅합니다.**

nginx 를 설치합니다

```
sudo apt-get install nginx
```

### SSL 인증서 발급받기

certbot을 활용하여 ec2 도메인에 대한 letsencrypt 인증서를 발급합니다.

```
sudo apt update
sudo apt-get install letsencrypt -y
cd /root
sudo service nginx stop
certbot --nginx
( 이후 SSL 인증서를 받을 도메인 입력 )
( redirect 2 설정 )
```

## nginx 를 통해 url 주소에 따라 백엔드 / 프론트엔드 요청을 처리합니다

/etc/nginx/sites-enabled/ 아래에 있는 \*.conf 파일을 생성 및 수정해서 사용자 설정을 구성합니다.

아래 코드는 예시입니다.

```
server {

    listen 8080;
    server_name hyodokebi.duckdns.org;
    return 301 https://localhost$request-uri;
}

server{
    listen 443 ssl;

    server_name hyodokebi.duckdns.org;

    ssl_certificate /etc/letsencrypt/live/hyodokebi.duckd
    ssl_certificate_key /etc/letsencrypt/live/hyodokebi.d

    location /{
        root /home/ubuntu/build;
        index index.html;
        try_files $uri /index.html;
    }
}
```

```

location /api {

    proxy_pass http://localhost:8081;
    charset utf-8;
    proxy_redirect off;

    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x
    proxy_set_header X-NginX-Proxy true;
    proxy_set_header X-Forwarded-Proto $scheme;

}
location /pyapi{

    proxy_pass http://localhost:5000;
    charset utf-8;
    proxy_redirect off;

    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x
    proxy_set_header X-NginX-Proxy true;
    proxy_set_header X-Forwarded-Proto $scheme;

}

}

```

## 프로젝트 파일을 EC2 내부에 저장 및 실행합니다

### GIT 에 있는 프로젝트를 EC2에 복제하기

```
git clone { git 주소 }
```

### SpringBoot 실행하기

1. spring 프로젝트의 설정파일인 application.yml 파일을 프로젝트의 resource 폴더로 복사
2. gradlew 를 통해 프로젝트 빌드 ( 이전에 권한을 줘야함 )
3. java -jar 명령어를 통해 build 폴더 내의 jar파일 실행

```
sudo cp ~/secret/application.yml ./src/main/resources/  
chmod +x ./gradlew  
./gradlew clean build  
java -jar ./build/libs/{@@}.jar
```

## React 실행하기

1. React 프로젝트의 src 폴더와 같은 레벨의 폴더 위치로 이동합니다.
2. 설정파일인 .env 파일을 추가합니다
3. 필요한 의존성을 설치 후 빌드합니다.

```
cp ~/secret/.env ~/  
npm install  
rm -rf build /*과거에 실행한적 있다면 실행 ( 현재 위치에 build 파일이  
npm run build  
  
sudo service nginx restart
```

## Nginx 실행하기

```
service nginx restart
```

## Flask 실행

1. flask 의 src로 이동합니다.
2. app을 실행합니다 ( 특별한 build 과정 필요 없음 )

```
pip install -r requirements.txt  
python3 app.py
```