포팅 매뉴얼

🗓️ 작성일 2023.10.05

👥 작성자 신준호

📄 제목 Book Wave 포팅 매뉴얼

💡 개발 환경

형상 관리

- Gitlab

이슈 관리

- Jira

Communication

- Mattermost
- Webex
- Notion

OS

- Windows 10

UI/UX

- Figma

IDE

- IntelliJ IDEA
- VSCode

Database

- MySQL 8.0.33
- Redis 7.0.12

BackEnd

- JAVA JDK 11
- SpringBoot 2.7.15
    - Gradle 8.2.1
    - Spring Data JPA
    - Lombok
    - Springdoc Swagger
    - Spring Security
    - JWT

DATA

- Python 3.9
- FastApi 0.103.1
- Selenium 4.13.0
- pandas 1.5.3

API

- 카카오로그인 API

FrontEnd

- Node 18.17.1
- NPM 9.6.7
- Vite + Typescript + SWC
- React 18.2.0
    - Recoil
    - React-query
    - Framer-motion
    - ChakraUI
    - Axios

- 설치
    - npm install react-router-dom
    - npm install --save-dev @types/react-router-dom
    - npm install --save-dev @types/react

기타 편의 툴

- Postman 10.15.8

Infra

- AWS EC2
    - Ubuntu 20.04
    - Docker 24

- Nginx 1.25.1
- Jenkins

💡 EC2 사용 포트

포트	설명
22	SSH
80	HTTP
443	HTTPS
3000	React
5012	MySQL
9090	Spring Boot
8000	FastApi
8080	Jenkins
6379	Redis

💡 Nginx 설정

/etc/nginx/sites-available/default 파일에 도메인 설정

```
server {
    listen 80;
    server_name j9b203.p.ssafy.io;
    ...
}
```

Nginx 리로드하여 default파일 설정 적용

```
# default 파일이 정상적인지 테스트
$ sudo nginx -t
```

```
# 리로드
$ sudo service nginx reload
```

SSL 인증서 발급(Certbot으로 Nginx에 SSL 인증서 자동 적용)

```
# Let's Encrypt 설치
$ sudo apt-get install letsencrypt

# Certbot 설치
sudo apt-get install certbot python3-certbot-nginx

# Certbot 동작
sudo certbot --nginx
```

SSL 인증서 적용 후 최종적으로 설정한 /etc/nginx/sites-available/default 파일

```
server {

        root /var/www/html;

        index index.html index.htm index.nginx-debian.html;

        server_name j9b203.p.ssafy.io;

        location / {
                proxy_pass http://localhost:3000;
        }

        location /api {
                proxy_pass http://localhost:9090;
        }
        location /rec {
                proxy_pass http://localhost:8000;
        }
        location /img {
                proxy_pass http://localhost:9090;
        }
        location /img/member{
                proxy_pass http://localhost:9090;
        }

    listen [::]:443 ssl ipv6only=on;
    listen 443 ssl;

    ssl_certificate /etc/letsencrypt/live/j9b203.p.ssafy.io/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/j9b203.p.ssafy.io/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    client_max_body_size 20M;
}

server {
    if ($host = j9b203.p.ssafy.io) {
        return 301 https://$host$request_uri;
    }

        listen 80 default_server;
        listen [::]:80 default_server;

        client_max_body_size 20M;

        server_name j9b203.p.ssafy.io;
    return 404;

}
```

💡 Docker 컨테이너

Jenkins

```
# jenkins 이미지 pull
$ sudo docker pull jenkins/jenkins:jdk11

# jenkins 컨테이너 실행
$ sudo docker run -d -p 8080:8080 -v /home/ubuntu/jenkins:/var/jenkins_home --name jenkins -u root jenkins/jenkins:jdk11
```

- EC2 서버의 /home/ubuntu/jenkins와 docker 컨테이너의 /var/jenkins_home를 볼륨으로 연결하여 workspace에 생성되는 프로젝트 빌드 파일을 관리

사용된 주요 플러그인

- Gradle Plugin 2.8.2
- NodeJS Plugin 1.6.1
- SSH Agent Plugin 333.v878b_53c89511
- GitLab Plugin 1.7.16

Item

- Item1: bookwave-be
  - Pipeline script

```
pipeline {
    agent any

    stages {
        stage('Git clone') {
            steps {
                git branch: 'master', credentialsId: 'jenkins_token', url: 'https://lab.ssafy.com/s09-bigdata-recom-sub2/S09P22B203.git'
            }
        }
        stage('Build') {
            steps {
                dir("./BE/bookwave") {
                    sh "chmod +x ./gradlew"
                    sh "./gradlew clean build"
                }
            }
        }
        stage('Deployment') {
            steps {
                sshagent(credentials: ['ssh_key']) {
                    sh '''
                        ssh -o StrictHostKeyChecking=no ubuntu@j9b203.p.ssafy.io
                        scp /var/jenkins_home/workspace/bookwave-BE/BE/bookwave/build/libs/bookwave-0.0.1-SNAPSHOT.jar ubuntu@j9b203.p.ssafy.io:/home/ubuntu/bookwave-back
                        ssh -t ubuntu@j9b203.p.ssafy.io ./deploy_be.sh
                    '''
                }
                timeout(time: 30, unit: 'SECONDS') {
                }
            }
        }
    }
}
```

- Item2: bookwave-fe
  - Pipeline script

```
pipeline {
    agent any

    stages {
        stage('Git clone') {
            steps {
                git branch: 'mast', credentialsId: 'jenkins_token', url: 'https://lab.ssafy.com/s09-bigdata-recom-sub2/S09P22B203'
                echo "Current workspace: ${workspace}"
            }
        }
        stage('Build') {
            steps {
                dir("./FE") {
                    nodejs(nodeJSInstallationName: 'NodeJS 18.17.1') {
                        // sh 'npm install && npm run build'
                        sh 'CI=false npm install && CI=false npm run build'
                    }
                }
            }
        }
        stage('Compression') {
            steps {
                dir("./FE") {
                    sh '''
                    rm -rf node_modules
                    tar -cvf bookwave_1.0.0.tar .
                    '''
                }
            }
        }
        stage('Deployment') {
            steps {
                sshagent(credentials: ['ssh_key']) {
                    sh '''
                        ssh -o StrictHostKeyChecking=no ubuntu@j9b203.p.ssafy.io "rm -rf /home/ubuntu/bookwave-front/node_modules"
                        scp -r /var/jenkins_home/workspace/bookwave-FE/FE/* ubuntu@j9b203.p.ssafy.io:/home/ubuntu/bookwave-front
                        ssh -t ubuntu@j9b203.p.ssafy.io ./deploy_fe.sh
                    '''
                }
            }
        }
    }
}
```

- Item3: bookwave-fastapi
    - Pipeline script

```
pipeline {
    agent any

    stages {
        stage('Git clone') {
            steps {
                git branch: 'master', credentialsId: 'jenkins_token', url: 'https://lab.ssafy.com/s09-bigdata-recom-sub2/S09P22B203.git'
            }
        }

        stage('Setup Python Environment') {
            steps {
                dir("./FASTAPI/bookwave") {
                    sh "python3 -m venv venv"
                    sh "source venv/bin/activate"
                    sh "pip install -r requirements.txt"
                }
            }
        }

        stage('Deployment') {
            steps {
                sshagent(credentials: ['ssh_key']) {
                    sh '''
                        scp -r S09P22B203 ubuntu@j9b203.p.ssafy.io:/home/ubuntu/fastapi_app
                        ssh -t ubuntu@j9b203.p.ssafy.io ./deploy_fastapi.sh
                    '''
                }
            }
        }
    }
}
```

Dockerfile & deploy.sh 생성

- Dockerfile
    - BE

```
FROM adoptopenjdk/openjdk11
COPY ./bookwave-0.0.1-SNAPSHOT.jar /bookwave-0.0.1-SNAPSHOT.jar
CMD ["java", "-jar", "bookwave-0.0.1-SNAPSHOT.jar"]
```

    - FE

```
FROM node:18.17.1-alpine

WORKDIR /frontend

COPY . ./

EXPOSE 3000

RUN npm install --silent

CMD ["npm", "run", "dev"]
```

  - FASTAPI

```
FROM python:3.9
WORKDIR /FASTAPI
COPY . ./
RUN pip install --no-cache-dir --upgrade -r requirements.txt
EXPOSE 8000
CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8000"]
```

- deploy.sh(소유자에게 파일 실행 권한 필요: -rwxrw-r—)
  - BE

```
cd bookwave-back
sudo docker compose down
sudo docker compose up -d --build
yes | sudo docker system prune
```

  - FE

```
cd bookwave-front
sudo docker compose down
sudo docker compose up -d --build

yes | sudo docker system prune
```

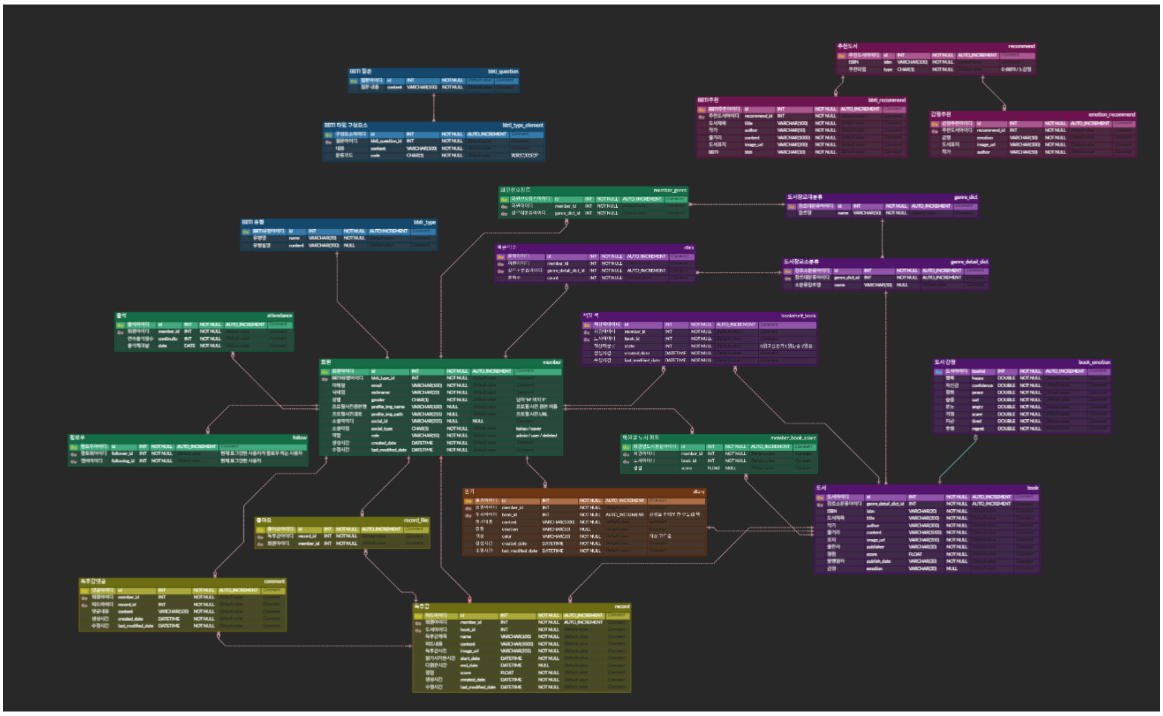
- FASTAPI

```
cd bookwave-fastapi
sudo docker compose down
sudo docker compose up -d --build

yes | sudo docker system prune
```

💡 MySQL 설정

ERD



운영 환경 계정

- database: bookwave
- host: j9b203.p.ssafy.io
- port: 5012
- username: bookwaveteam203

    - password: bwteam203

💡 Redis 설정

- port: 6379
- password: redis

💡 Spring Boot properties 파일

9.1 application.properties

```
spring.profiles.active=prd

# jpa
spring.jpa.show-sql=true
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
spring.jpa.properties.hibernate.format_sql=true
logging.level.org.hibernate.type.descriptor.sql=trace

# server port
server.port=9090

# swagger
springdoc.swagger-ui.path=/swagger-ui

# oauth kakao
oauth2.kakao.client-id=d162981d38b2d120dd8de1714dd68c63

# oauth naver
oauth2.naver.client-id=aWQgyiORSXixtGdZLzfh
oauth2.naver.client-secret=fIvbgxHXcK
oauth2.naver.state=test

# JWT
jwt.secretKey=aWQgyiORSXixtGdZLzfh
#jwt.access.expiration=3600000
jwt.access.expiration=3600000
jwt.access.header=Authorization
jwt.refresh.expiration=18000000
jwt.refresh.header=Authorization-refresh
```

```
# hibernate_sequence ??? ?? ???
spring.jpa.hibernate.use-new-id-generator-mappings=false
```

9.1 application-dev.properties

```
spring.datasource.url=jdbc:mysql://localhost:3306/bookwave?serverTimezone=Asia/Seoul&c
haracterEncoding=UTF-8
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.show_sql=false

redis.host=localhost
redis.port=6379
redis.password=redis

#oauth2.kakao.redirect-uri=https://j9b203.p.ssafy.io/login/oauth2/code/kakao
oauth2.kakao.redirect-uri=http://localhost:3000/login/oauth2/code/kakao


file.profile.path=
host.profile.url=http://localhost:9090/

file.member-upload-dir=/bookwave/upload/images/member/
file.member-server-domain=http://localhost:9090
file.member-url-path=/img/

file.record-upload-dir=/bookwave/upload/images/record/
file.record-server-domain=http://localhost:9090
file.record-url-path=/img/

spring.servlet.multipart.max-file-size=10MB
spring.servlet.multipart.max-request-size=100MB
```

9.3 application-prd.properties

```
spring.datasource.url=jdbc:mysql://j9b203.p.ssafy.io:5012/bookwave?serverTimezone=Asi
a/Seoul&useUnicode=yes&characterEncoding=UTF-8
spring.datasource.username=bookwaveteam203
spring.datasource.password=bwteam203
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver


spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.show_sql=false
```

```
redis.host=j9b203.p.ssafy.io
redis.port=6379
redis.password=redis

#oauth2.kakao.redirect-uri=https://j9b203.p.ssafy.io/login/oauth2/code/kakao
oauth2.kakao.redirect-uri=http://localhost:3000/login/oauth2/code/kakao


file.profile.path=
host.profile.url=https://j9b203.p.ssafy.io/

file.member-upload-dir=/bookwave/upload/images/member/
file.member-server-domain=https://j9b203.p.ssafy.io/img/member/
file.member-url-path=/img/

file.record-upload-dir=/bookwave/upload/images/record/
file.record-server-domain=https://j9b203.p.ssafy.io
file.record-url-path=/img/


spring.servlet.multipart.max-file-size=20MB
spring.servlet.multipart.max-request-size=100MB
```

💡 React 프로젝트 환경변수(.env 파일)

로컬 환경에서의 .env

```
VITE_REACT_APP_KAKAO_REDIRECT_URL=http://localhost:3000/login/oauth2/code/kakao
```

운영 환경에서의 .env

```
VITE_REACT_APP_KAKAO_REDIRECT_URL=https://j9b203.p.ssafy.io/login/oauth2/code/kakao
```

💡 소셜 로그인 Redirect URI

Kakao

- https://j9b203.p.ssafy.io/login/oauth2/code/kakao
- http://localhost:3000/login/oauth2/code/kakao