

추천 못하면 죽는 병 걸림

Porting Manual



I. 빌드 및 배포

1. 개발 환경

Server : AWS EC2 Ubuntu 22.04.1 LTS

Visual Studio Code : 1.72.1

IntelliJ IDEA : 2022.1.4(Ultimate Edition)11.0.15 + 10-b2043.56 amd64

JVM : 11.0.16 + 8-post-Ubuntu-Oubuntu120.04

Docker : 20.10.18

MySQL : 8.0.28

NginX : 1.23.1

Jenkins : 1.18.0-Oubuntu1.3

Node.js : 16.14.2

Vmware Workstation : 16.2.4

Hadoop : 3.2.2

2. 설정 파일 및 환경 변수

React

- DockerFile

```
FROM nginx:stable-alpine as production-stage
COPY ./nginx/nginx.conf /etc/nginx/conf.d/default.conf
COPY ./pem /pem
COPY ./build /usr/share/nginx/html
EXPOSE 80
CMD ["nginx","-g","daemon off;"]
```

- ExcuteShell

```
#Jenkins
echo "프론트 빌드 시작"
pwd
cd frontend
npm install
CI = false npm run build
# Jenkins 내부의 Docker 관련 설정들
sudo docker stop reactapp
# reactapp 삭제
sudo docker rm reactapp
# reactapp 이미지 파일 build
sudo docker build -t reactapp
# 80포트와 443포트(https 적용 시 443, ssh 도 열어줘야한다)를 열고 reactapp 컨테이너 실행.
sudo docker run -i -t -d -p 80:80 -p 443:443 --name reactapp reactapp
```

2. 설정 파일 및 환경 변수

Spring

- application.properties

Data Source

```
spring.datasource.url=jdbc:mysql://{DB container name}:3306/{schema name}? useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Seoul&zeroDateTimeBehavior=convertToNull&rewriteBatchedStatements=true
```

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

```
spring.datasource.hikari.username=a207
```

```
spring.datasource.hikari.password=a207 계정 password
```

#JPA

```
spring.jpa.hibernate.naming.implicit-strategy=org.springframework.boot.orm.jpa.hibernate.SpringImplicitNamingStrategy
```

```
spring.jpa.hibernate.ddl-auto=update
```

```
spring.jpa.generate-ddl=true
```

```
spring.jpa.show-sql=true
```

```
spring.jpa.properties.hibernate.format_sql=true
```

```
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL57Dialect
```

```
spring.jpa.properties.hibernate.format_sql=true
```

```
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL57Dialect
```

```
logging.level.root=INFO
```

2. 설정 파일 및 환경 변수

- deploy.sh

```
echo '실행 시작'
echo 'git pull'
echo 'jar 파일 삭제'
rm build/libs/*.jar
echo '빌드 시작'
./gradlew build
```

```
echo '도커파일 이미지 빌드'
docker build -t springbootapp .
echo '컨테이너 중지'
docker stop springbootapp
echo '컨테이너 삭제'
docker rm springbootapp
echo '컨테이너 실행'
```

8000포트 열어서 springbootapp 이라는 이름의 컨테이너로 빌드

```
docker run -p 8000:8000 --name springbootapp --network ubuntu_defau
lt -d springbootapp
```

- DockerFile

```
FROM openjdk:8-jdk
ARG JAR_FILE = build/libs/*.jar
COPY ${JAR_FILE} app.jar
ENTRYPOINT ["java","-jar","-Duser.timezone = Asia/Seoul","/app.jar"]
```

2. 설정 파일 및 환경 변수

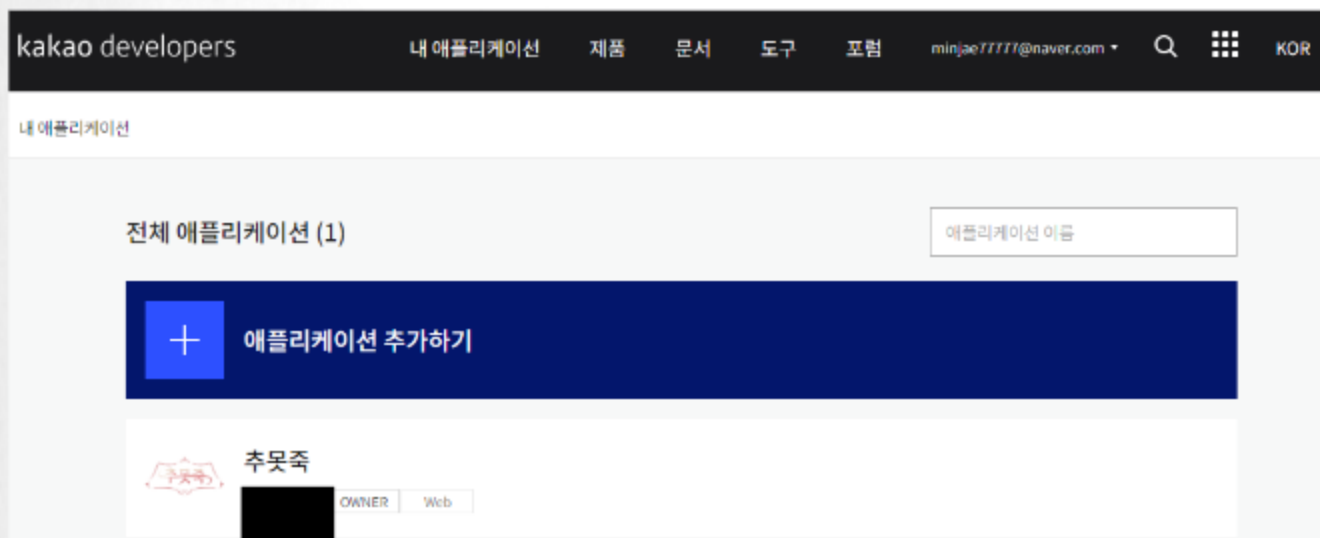
- ExecuteShell

```
# deploy.sh에 적힌 명령어 수행
cd backend
chmod +x gradlew
chmod +x deploy.sh
./deploy.sh
```

3. 외부 서비스

-Kakao

1. <https://developers.kakao.com/>에 접속해서 애플리케이션을 추가한다
(내 애플리케이션 > 애플리케이션 추가하기 > 앱 이름 입력 > 사업자 명 입력 > 저장)



3. 외부 서비스

2. 내 애플리케이션에서 앱 키를 확인하고 기록한다.

(좌측 Nav 바에서

앱 설정 > 요약 정보 > 앱 키 > REST API키

제품 설정 > 카카오 로그인 > 보안 > Client Secret의 코드 발급)

kakao developers

내 애플리케이션 제품 문서 도구 포럼 minjae77777@naver.com

내 애플리케이션 > 앱 설정 > 앱 키

추천

ID 831214 OWNER Web

앱 키

종류	앱 키	재발급
네이티브 앱 키		보시 재발급
REST API 키		보시 재발급
JavaScript 키		보시 재발급
Admin 키		보시 재발급

- 네이티브 앱 키: Android, iOS SDK에서 API를 호출할 때 사용합니다.
- JavaScript 키: JavaScript SDK에서 API를 호출할 때 사용합니다.
- REST API 키: REST API를 호출할 때 사용합니다.
- Admin 키: 모든 권한을 갖고 있는 키입니다. 노출이 되지 않도록 주의가 필요합니다.

3. 사이트 도메인을 등록(로그인 사용 시 Redirect URI)한다.

kakao developers

내 애플리케이션 제품 문서 도구 포럼 minjae77777@naver.com

내 애플리케이션 > 앱 설정 > 플랫폼

Android

Android 플랫폼 등록

iOS

iOS 플랫폼 등록

Web

신세 수정

사이트 도메인 http://localhost:3000

- 가기로 로그인 시점 시 Redirect URI를 등록해야 합니다. [등록하기](#)

3. 외부 서비스

4. 로그인 동의 항목 구성을 설정한다.

(제품 설정 > 카카오 로그인 > 동의항목에서 위처럼 설정)

kakao developers
내 애플리케이션
제품
문서
도구
포럼
minjae77777@naver.com

내 애플리케이션 > 제품 설정 > 카카오 로그인 > 동의항목

비즈니스 설정 바로가기

개인정보

항목 이름	ID	상태	
닉네임	profile_nickname	● 필수 동의	설정
프로필 사진	profile_image	● 필수 동의	설정
카카오계정(이메일)	account_email	● 선택 동의	설정
성별	gender	● 선택 동의	설정
연령대	age_range	● 선택 동의	설정
생일	birthday	● 사정 면한	설정
출생 연도	birthyear	○ 권한 없음	

5. 로그아웃 사용 시 사용 할 Redirect URI 을 등록한다.

내 애플리케이션 > 제품 설정 > 카카오 로그인 > 고급

추천주
ID 821274 OAuth Web

카카오 로그인 ON

Logout Redirect URI
삭제 수정

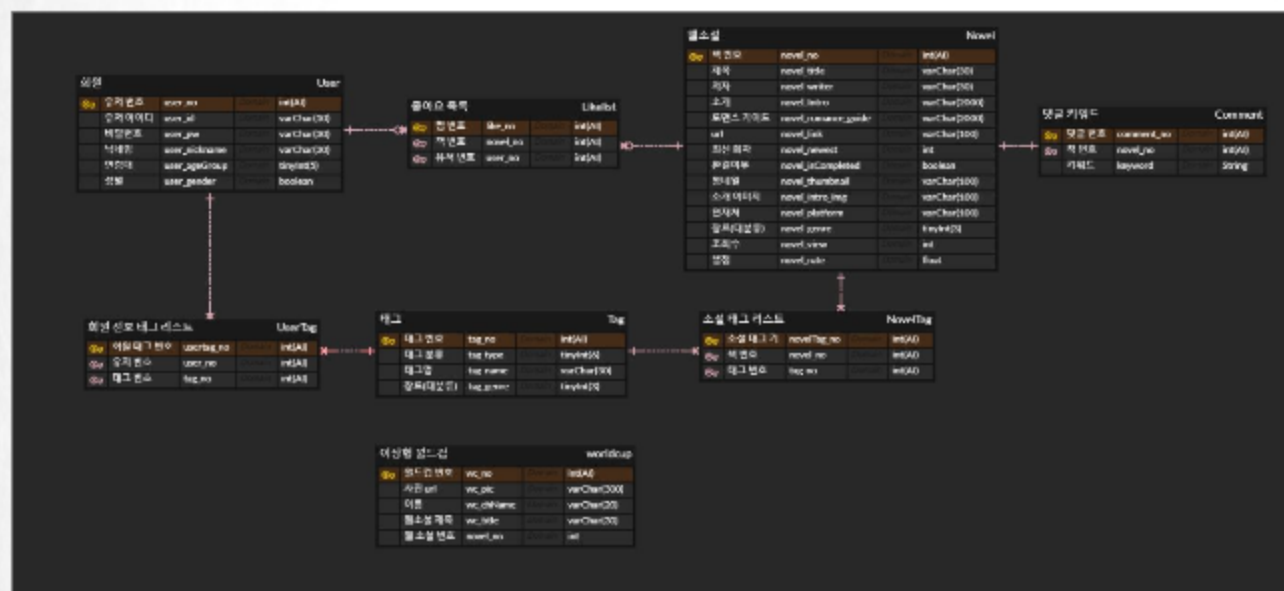
Logout Redirect URI	http://localhost:3000/logout
---------------------	------------------------------

- 카카오 로그인에서 사용한 Logout Redirect URI를 분량입니다. (최대 10개)
- 서비스 로그아웃 시 카카오 계정과 함께 로그아웃을 위한 소켓으로 선택할 수 있습니다.



II. 설계

1. ERD Cloud



PORTING MANUAL

추천 못하면 죽는 병 걸림