

# 빌드 및 배포 정리

서울 5반 5조

## 1) 프로젝트 기술 버전(툴, 버전 O)

개발 환경

Backend

- Java : jdk11 (11.0.17)
- gradle : 7.6
- IntelliJ : 2022.3.2
- MySQL : 5.7.39

Fronted

- react : 8.19.3
- node.js : 16.19.1
- VS Code : 1.74.3

AWS EC2

- ubuntu : 20.04.5
- Docker : 20.10.12
- Nginx : 1.18.0

Big Data

- Python : 3.8.2

OS

- Windows 10

Cooperation & Communication

- Gitlab
- Jira
- MatterMost
- Notion
- Discord

## 2) 빌드 방식

빌드 방식:

Frontend

```
/frontend  
npm i  
npm run dev
```

Backend

```
/Backend/billboard
```

```
./gradlew clean bootJar 실행
```

```
java -jar -Dspring.profiles.active=local build/libs/*.jar 혹은 루트에서 ./gradlew bootRun
```

Fastapi

```
source venv/Scripts/activate  
pip install -r requirements.txt  
uvicorn main:app --reload
```

## 3) 배포 방법 정리

### fastapi 배포 - dockerfile

---

```
FROM python:3.8.2  
  
WORKDIR /app/  
  
COPY . /app/  
COPY ./requirements.txt /app/  
  
RUN pip install -r requirements.txt  
  
CMD uvicorn --host=0.0.0.0 --port 8000 main:app
```

---

## 프론트엔드 빌드 및 배포

---

- Dockerfile

```
• FROM nginx:stable-alpine
•
• WORKDIR /app
•
• RUN mkdir ./dist
•
• ADD ./dist ./dist
•
• RUN rm /etc/nginx/conf.d/default.conf
•
• COPY ./nginx.conf /etc/nginx/conf.d
•
• EXPOSE 3000
•
• CMD ["nginx", "-g", "daemon off;"]
```

- 배포

# 프론트엔드 배포 /fronted

cd frontend/billboard

docker stop react

docker rm react

npm i

npm run build

docker build -t react .

docker run -d -p 3000:3000 --name react react백엔드 빌드 및 배포

## 백엔드 빌드 및 배포

---

- Dockerfile

```
FROM openjdk:11-jdk-slim
ARG JAR_FILE=build/libs/billboard-0.0.1-SNAPSHOT.jar
COPY ${JAR_FILE} app.jar
ENV JAVA_OPTS=""
ENTRYPOINT ["java", "-jar", "/app.jar"]
```

- 배포

# 백엔드 배포 /backend/billboard

```
docker stop spring
docker rm spring
gradle clean build
docker build -t spring .
docker run -d -p 8081:8081 --name spring spring
```

### MYSQL 배포(Docker)

---

```
sudo docker pull mysql:5.7.39

sudo docker images

sudo docker run --name mysql-container -e MYSQL_ROOT_PASSWORD=root
-d -p 3306:3306 mysql:5.7.39

sudo docker exec -it mysql-container bash

mysql -u root -p

Enter password : root

create user '계정이름'@'localhost' identified by '비밀번호';

grant all privileges on *.* to '사용자'@'localhost' identified by '비밀번호';

flush privileges;

show grants for 'ssafy'@'%';

#Ctrl + P + Q
```

4) ERD, properties

Gradle 의존성:

```
plugins {
    id 'java'
    id 'org.springframework.boot' version '2.7.8'
    id 'io.spring.dependency-management' version '1.0.15.RELEASE'
}

group = 'com.ssafy'
version = '0.0.1-SNAPSHOT'
sourceCompatibility = '11'

configurations {
    compileOnly {
        extendsFrom annotationProcessor
    }
}

repositories {
    mavenCentral()
}

dependencies {
```

```

// spring boot
implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'
implementation 'org.springframework.boot:spring-boot-starter-web'
developmentOnly 'org.springframework.boot:spring-boot-devtools'
testImplementation 'org.springframework.boot:spring-boot-starter-test'

// jpa
implementation 'org.springframework.boot:spring-boot-starter-data-jpa'

// spring security
implementation 'org.springframework.boot:spring-boot-starter-security'

// lombok
compileOnly 'org.projectlombok:lombok'
annotationProcessor 'org.projectlombok:lombok'

// mysql
runtimeOnly 'mysql:mysql-connector-java:8.0.27'

// jwt
implementation 'io.jsonwebtoken:jjwt-api:0.11.2'
implementation 'io.jsonwebtoken:jjwt-impl:0.11.2'
implementation 'io.jsonwebtoken:jjwt-jackson:0.11.2'

// com.sun.xml.bind
implementation 'com.sun.xml.bind:jaxb-impl:4.0.1'
implementation 'com.sun.xml.bind:jaxb-core:4.0.1'

// javax.xml.bind
implementation 'javax.xml.bind:jaxb-api:2.4.0-b180830.0359'

// email 인증// https://mvnrepository.com/artifact/com.sun.activation/javax.activation
implementation group: 'com.sun.activation', name: 'javax.activation', version: '1.2.0'
implementation 'org.springframework.boot:spring-boot-starter-mail:2.7.0'

// swagger
implementation group: 'org.springdoc', name: 'springdoc-openapi-ui', version: '1.6.9'

implementation 'com.fasterxml.jackson.core:jackson-databind:2.13.0'
}

tasks.named('test') {
    useJUnitPlatform()
}

tasks {
    processResources {
        duplicatesStrategy = org.gradle.api.file.DuplicatesStrategy.INCLUDE
    }
}

```

## Spring boot Application properties:

```
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp

spring.datasource.url=jdbc:mysql://j8a505.p.ssafy.io:3306/billboard?serverTimezone=UTC&rewriteBatchedStatements=true
spring.datasource.username=ssafy
spring.datasource.password=ssafy

spring.mvc.pathmatch.matching-strategy = ANT_PATH_MATCHER
server.port=8081

logging.level.xyz.quietjun=trace

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLInnoDBDialect
spring.jpa.hibernate.ddl-auto=none
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
spring.jpa.properties.hibernate.jdbc.batch_size=500

spring.mail.host=smtp.gmail.com
spring.mail.port=465
spring.mail.username=wjsjj505@gmail.com
spring.mail.password=zvvdasjiwobgeipi
spring.mail.properties.debug=true
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.ssl.enable=true
spring.mail.properties.mail.smtp.starttls.enable=true
spring.mail.properties.mail.smtp.ssl.trust=smtp.gmail.com

jwt.secret-key=VlwEyVBsYt9V7zq57TejMnVUyzbIYcfPQye08f7MGVA9XkHa
```

## React package.json:

```
{
  "name": "billboard",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  "start": "HOST=127.0.0.1 react-scripts start",
  "scripts": {
    "dev": "vite --port 3000",
    "build": "vite build",
    "preview": "vite preview",
    "lint": "eslint src --ext .jsx",
    "lint:fix": "npm run lint -- --fix"
  },
  "dependencies": {
    "@ant-design/icons": "^5.0.1",
    "@emotion/react": "^11.10.6",
    "@emotion/styled": "^11.10.6",
    "@mui/icons-material": "^5.11.11",
    "@mui/joy": "^5.0.0-alpha.73",
    "@mui/material": "^5.11.13",
    "@reduxjs/toolkit": "^1.9.3",
    "antd": "^5.3.1",
    "axios": "^1.3.4",
    "dotenv": "^16.0.3",
    "fetch-jsonp": "^1.2.3",
    "html-webpack-plugin": "^5.5.0",
```

1. *Journal of Management Studies*, 1990, 27, 1, 1-14.



## API 명세서

구분	base_uri 기능	URI <a href="http://88a505.p.ssafy.io">http://88a505.p.ssafy.io</a>	메소드	Request Body	Response	상태 코드	비고	구현 완료	접근 권한	
	회원정보등록	/api/users	POST	{ userid : 아이디(String) password : 비밀번호(String) nickname : 닉네임(String) email : 이메일(String) }		201 성공 400 BR 409 중복된 아이디/이메일		<input checked="" type="checkbox"/>	ANY	이메일 인증 여부
	회원정보수정	/api/users/{userid}	PUT	{ password : 비밀번호(String) nickname : 닉네임(String) email : 이메일(String) }		200 성공 404 존재하지 않는 아이디	이메일 변경 시 재인증 필요 - 사회질서 예방(안 할 예정)	<input type="checkbox"/>	SELF	
	회원정보삭제	/api/users/{userid}	DELETE			200 성공 404 존재하지 않는 아이디		<input checked="" type="checkbox"/>	SELF	
	로그인	/api/users/login	POST	{ userid : 아이디(String) password : 비밀번호(String) }	{ accessToken : token(String), refreshToken : token(String), userInfo : { nickname : 닉네임(String), } }	200 성공 400 BR 401 실패	FE에서 필요한 사항 요청 - 여기에서 accessToken도 받아올 필요	<input checked="" type="checkbox"/>	ANY	로그인 필요
	로그아웃	/api/users/logout/{userid}	POST			200 성공 401 실패 404 ID 못찾음		<input checked="" type="checkbox"/>	SELF	
	ID 중복 확인	/api/users/check-id/{userid}	GET			200 성공가능 409 중복된 아이디		<input checked="" type="checkbox"/>	ANY	
	이메일 인증	/api/users/email-auth	POST	{ email : 이메일(String) }		200 인증 성공 409 사용중인 email	이메일 인증 어떻게 하는 지 따라서 상태코드랑 토직 변경	<input checked="" type="checkbox"/>	ANY	
	인증번호 확인	/api/users/check-authkey	POST	{ email : 이메일(String) authKey : 인증번호(String) }		200 인증 성공 400 BR 401 일치하지 않음 404 일치하지 않음 email 410 인증된 인증	-> 임시 비밀번호로 이메일로 보냄	<input checked="" type="checkbox"/>	ANY	인증 완료/만료 후
	ID 찾기	/api/users/find-id/{email}	GET			200 인증 성공 404 중복되지 않은 email		<input checked="" type="checkbox"/>	ANY	

## 와이어프레임

