

# 포팅 메뉴얼

## 사용 기술 스택

front-end : NodeJS, React

back-end: Java, Spring boot, JPA, gradle, Docker, NginX, Jenkins

기술 스택	버전
NodeJS	19.0.0
React	18.2.0
OpenJDK	1.8.0_192
Spring boot	2.4.5
QueryDSL	4.4.0
Docker	20.10.17
NginX	1.18.0
Jenkins	2.346
Gradle	6.9

## Frontend 배포

### 로컬 실행 방법

```
git clone https://lab.ssafy.com/s07-final/S07P31E106.git
cd frontend
npm i
npm start
```

### Dockerfile 을 사용한 빌드 및 배포

Dockerfile을 작성하여 Nginx와 react를 함께 배포합니다.

프로젝트내의 frontend 디렉토리의 루트 경로에서 다음 명령어를 실행합니다.

S07P31E106/frontend/nginx.conf

```
server {
    listen 80;
```

```

    location / {
        root    /app/build;
        index   index.html;
        try_files $uri $uri/ /index.html;
    }
}

```

- nginx 경로를 설정할 nginx.conf 파일을 추가합니다

S07P31E106/frontend/Dockerfile

```

# Dockerfile

# nginx 이미지를 사용합니다. 뒤에 tag가 없으면 latest 를 사용합니다.
FROM nginx

# root 에 app 폴더를 생성
RUN mkdir /app

# work dir 고정
WORKDIR /app

# work dir 에 build 폴더 생성 /app/build
RUN mkdir ./build

# host pc의 현재경로의 build 폴더를 workdir 의 build 폴더로 복사
ADD ./build ./build

# nginx 의 default.conf 를 삭제
RUN rm /etc/nginx/conf.d/default.conf

# host pc 의 nginx.conf 를 아래 경로에 복사
COPY ./nginx.conf /etc/nginx/conf.d

# 80 포트 오픈
EXPOSE 80
# https 사용을 위한 443 포트 오픈
EXPOSE 443
# container 실행 시 자동으로 실행할 command. nginx 시작함
CMD ["nginx", "-g", "daemon off;"]

```

그 다음 도커파일을 작성합니다 nginx.conf 에 적혀있는 것을 컨테이너내 설정으로 변경합니다.

이후 다음 명령어를 차례로 입력하여 module을 설치, 빌드, 도커 이미지를 만들고 컨테이너에서 배포합니다.

```

# module 설치
npm update

# 빌드 파일 생성

```

```
CI=false npm run build

# 도커 이미지 빌드
docker build -t front:0.1 .

# 도커 컨테이너를 이용한 프론트엔드 배포
docker run --name front -d -p 3000:80 front:0.1
```

## backend 배포

### 로컬에서 실행 방법

build 결과물 얻기

```
git clone https://lab.ssafy.com/s07-final/S07P31E106.git
cd backend
```

S07P31E106/backend 에서

```
gradle clean build
```

그러면 .jar 파일이 S07P31E106/backend/build/libs 위치에 생성됩니다.

```
java -jar [jar파일명]
```

을 실행하면 자바 백엔드 서버가 열립니다.

### Dockerfile을 사용한 배포( 추후 자동 배포를 위한 과정)

Dockerfile을 backend 폴더에서 작성합니다.

S07P31E106/backend의 Dockerfile

```
FROM openjdk:8
VOLUME /tmp
EXPOSE 8080
ARG JAR_FILE=build/libs/ssafy-web-project-1.0-SNAPSHOT.jar
COPY ${JAR_FILE} app.jar
ENTRYPOINT ["java", "-jar", "/app.jar"]
ARG DEBIAN_FRONTEND=noninteractive
```

```
ENV TZ=Asia/Seoul  
RUN apt-get install -y tzdata
```

## S07P31E106/backend

```
docker build -t back:0.1 .  
docker run --name back -p 8081:8080 back:0.1
```

다음을 실행하면 배포가 완료됩니다.