포팅 매뉴얼

프로젝트 기술 스택

Vue	2
Vuetify	2.6.9
JAVA	11
SpringBoot	2.7.1
JPA	
QueryDSL	
Spring security	
MySQL	8.0.30
Redis	latest (alpine3.16 다음 버전)
Docker	20.10.17
Jenkins	2.346.3
JWT	0.11.5
OAuth	2.0
Nginx	latest (stable-alpine-perl 다음 버전)

프로젝트 환경 설정

Kakao Developers 설정

1. **애플리케이션 등록**

내 애플리케이션 > 애플리케이션 추가하기

-“prosn”

2. **도메인 등록**

   내 애플리케이션 > 앱 설정 > 플랫폼 > Web

3. **Redirect URI 설정**

   내 애플리케이션 > 제품 설정 > 로그인

   -Redirect URI 추가

4. **카카오 로그인 활성화**

   내 애플리케이션 > 제품 설정 > 로그인

   -활성화 설정 ON

5. **인가 코드 받기**

   *카카오 로그인 시, 토큰을 받고 해당 토큰으로 사용자 정보를 받기 위해 인가 코드가 필요함

   내 애플리케이션 > 앱 설정 > 앱 키 (REST API 키)

   GET

   HTTP/1.1

   kauth.kakao.com

   oauth/authorize?client_id={CLIENT_ID}&redirect_uri={REDIRECT_URI}&response_type=code

6. **토큰 받기**

   POST

   HTTP/1.1

   kauth.kakao.com

   oauth/token

   Content-type: application/x-www-form-uriencodedcharset=utf-8

7. **사용자 정보 받기**

   POST

   HTTP/1.1

   kapi.kakao.com

v2/user/me

Content-type: application/x-www-form-urlencoded.charset=utf-8

Naver Developers 설정

1. **애플리케이션 등록**

   애플리케이션 등록

   -“prosn”

2. **도메인 등록**

   내 애플리케이션 > API 설정 > 로그인 오픈 API서비스 환경

   -서비스 URL 등록

3. **Redirect URI 설정**

   내 애플리케이션 > API 설정 > 로그인 오픈 API서비스 환경

   -Callback URL

4. **인가 코드 받기**

*네이버 로그인 시, 토큰을 받고 해당 토큰으로 사용자 정보를 받기 위해 인가 코드와 “state”라는 변수가 필요함

내 애플리케이션 > 개요 > 애플리케이션 정보 (Client ID)

GET

HTTP/1.1

nid.naver.com

oauth2.0/authorize?response_type=code&client_id={CLIENT_ID}&state=state&redirect_uri={CALLBACK_URL}

5. **토큰 받기**

   POST

   HTTP/1.1

   nid.naver.com

   oauth2.0/token

   Content-type: application/x-www-form-urlencoded.charset=utf-8

6. **사용자 정보 받기**

   POST

HTTP/1.1

openapi.naver.com

v1/nid/me

Content-type: application/x-www-form-urlencoded.charset=utf-8

EC2 세팅

- Docker 설치
- MySQL 설치
- Redis 설치
  - redis.conf 파일 내에서 호스트 localhost를 서버 도메인으로 변경

MySQL 유저 및 데이터베이스

username `prosn`

userpassword `prosn705@#`

DB table name `prosn;`

Nginx Default 값

nginx.conf

파일 위치: frontend 폴더 내에 위치

```
server {
    listen 80;

    location / {
        alias /usr/share/nginx/html/;
        try_files $uri $uri/ /index.html;
    }
}
```

Dockerfile

FE

파일 이름: Dokcerfile

파일 위치: frontend 빌드 파일 폴더 내에 위치

```
FROM nginx

RUN rm /etc/nginx/conf.d/default.conf
COPY ./nginx.conf /etc/nginx/conf.d/nginx.conf

COPY ./dist /usr/share/nginx/html

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]
```

BE

파일 이름: Dockerfile

파일 위치: backend 빌드 파일 폴더 내에 위치

```
FROM openjdk:11

EXPOSE 8080

ARG JAR_FILE=./build/libs/prosn-0.0.1-SNAPSHOT.jar

COPY ${JAR_FILE} prosn.jar

ENTRYPOINT ["java", "-jar", "prosn.jar"]

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

도커 이미지화 명령어

`docker build -t {빌드파일이름} {Dockerfile_있는_폴더위치}`

`ex) docker build -t prosnfrontend ./`

`ex) docker build -t prosnbackend ./`

배포 명령어 정리

FE `docker run -d -p 8081:80 prosnfrontend`

BE `docker run -d -p 8080:8080 prosnbackend`

MySQL

`docker run --name mysql-container -e MYSQL_ROOT_PASSWORD=<password> -d -p 3306:3306 mysql:8.0.30`

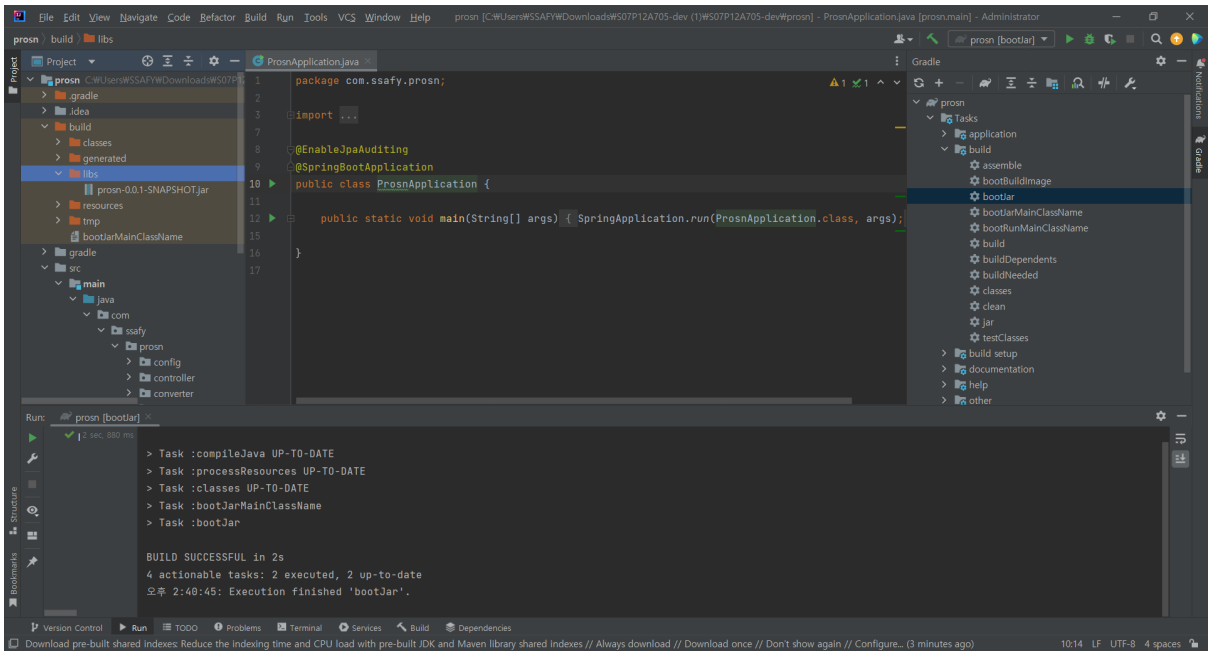
Redis

`docker run -p 6379:6379 redis`

프로젝트 빌드

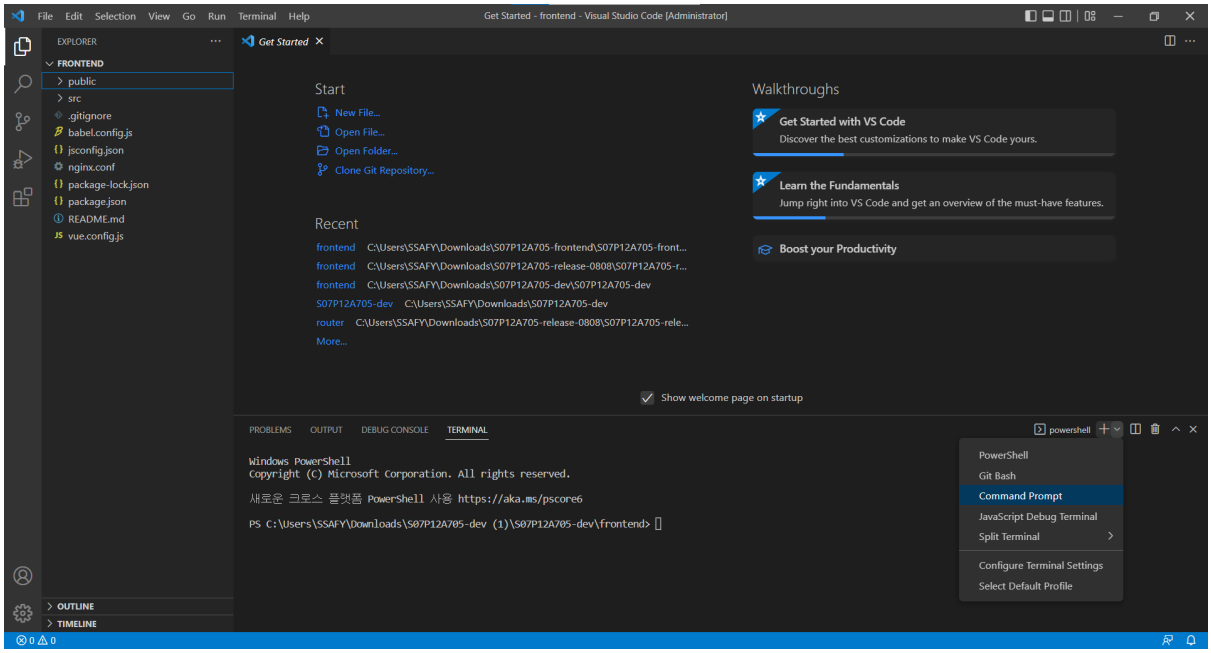
백엔드 빌드 방법

1. IntelliJ IDE로 프로젝트 임포트
2. 오른쪽 사이드 바에서 Gradle 탭 → Tasks → build → bootJar 클릭
3. 왼쪽 Project Explorer 에서 build 폴더 → libs → prosn-0.0.1-SNAPSHOT.jar



프론트엔드 빌드 방법

1. VScode로 프로젝트 임포트
2. 상단 메뉴바에서 Terminal → new Terminal
3. Command Prompt 켜기



4. 프롬프트 창에 `npm install`
5. `npm run build`
6. 왼쪽 Project explorer에 dist 파일이 빌드 파일