

**당신의 목소리를 들려주세요! HYPE VOICE**

**포팅 매뉴얼**

**목차**

***I. 개요***

**1. 프로젝트 개요**

**2. 개발환경**

**3. 프로젝트 사용 도구**

**4. 외부 서비스**

***II. 빌드***

**1. 환경변수**

**2. 빌드 방법**

**3. 배포**

**4. 서비스 이용 방법**

**I. 개요**

**1. 프로젝트 개요**

최근 게임, 애니메이션, 영화 등 많은 분야에서 필요로 하는 성우라는 직업에 대해 아시나요?

이렇게 많은 분야에서 필요로 하는 인기있는 성우가 되기 위한 과정은 쉽지 않습니다.

하지만 HYPE VOICE와 함께라면 그런 걱정은 안녕~

성우가 되고 싶은 당신! 그런 당신을 위한 HYPE VOICE 가 있습니다!

**2. 개발 환경**

- IntelliJ : 2023.3.4

- Java : 17.0.9

- Springboot : 3.2.2

- Spring Data JPA : 3.2.2

- Spring Security : 6.2.1

- QueryDSL : 5.0.0

- JPQL : 6.4.1

- Junit : 5.10.1

- Visual Studio Code : 1.85.1

- React : 2.27.0

- Typescript : 5.2.2

- Recoil : 0.7.7

- Tanstack-React-Query : 5.17.19

- DB : MySQL

- Server : AWS EC2

**3. 프로젝트 사용 도구**

이슈 관리 : Jira

형상 관리 및 코드 리뷰 : Gerrit

커뮤니케이션 : Mattermost, Notion

디자인 : Figma

배포 : Docker

**4. 외부 서비스**

- AWS S3

- OpenVidu

- Kakao OAuth

- Naver OAuth

**II. 빌드**

**1. 환경변수 형태**

application.yml

*# server:*  
 port: 포트 번호  
  
 *# datasource:*  
 driver-class-name: com.mysql.cj.jdbc.Driver  
 url: MySQL DB 주소

username: 유저 이름  
 password: 유저 패스워드

application-secret.yml

*# ec2 MySQL*EC2\_HOST: hypevoice.site  
EC2\_USERNAME: EC2 유저 이름  
EC2\_PASSWORD: EC2 패스워드  
  
*# jwt*jwt:  
 secret:  
 key: jwt 키  
  
*# aws s3*cloud:  
 aws:  
 credentials:  
 accessKey: aws 엑세스 키  
 secretKey: aws 시크릿 키

s3:  
 bucket: hypevoice-bucket  
 region:  
 static: ap-northeast-2  
 stack:  
 auto: false  
  
*# openvidu*OPENVIDU\_URL: https://demos.openvidu.io/  
OPENVIDU\_SECRET: openvidu 시크릿키

*# oauth*spring:  
 security:  
 oauth2:  
 client:  
 registration:  
 naver:  
 client-id: 사용자 아이디

client-secret: 사용자 비밀키  
 redirect-uri: "{baseUrl}/{action}/oauth2/code/{registrationId}"  
 authorization-grant-type: authorization\_code  
 scope: name, email *# 동의 항목* client-name: Naver  
 kakao:  
 client-id: 사용자 아이디client-secret: 사용자 아이디  
 redirect-uri: "{baseUrl}/{action}/oauth2/code/{registrationId}"  
 client-authentication-method: client\_secret\_post  
 authorization-grant-type: authorization\_code  
 scope: profile\_nickname, account\_email *# 동의 항목* client-name: Kakao  
 provider:  
 naver:  
 authorization\_uri: https://nid.naver.com/oauth2.0/authorize  
 token\_uri: https://nid.naver.com/oauth2.0/token  
 user-info-uri: https://openapi.naver.com/v1/nid/me  
 user\_name\_attribute: response  
 kakao:  
 authorization-uri: https://kauth.kakao.com/oauth/authorize  
 token-uri: https://kauth.kakao.com/oauth/token  
 user-info-uri: https://kapi.kakao.com/v2/user/me  
 user-name-attribute: id

**2. 빌드하기**

1) Front

npm i

npm create vite@latest

npm run dev

2) Back

Gradle 실행

build/libs/ 에 있는 jar 파일 실행

**3. 배포하기**

**-도커 이미지 배포(스프링 부트)**

**도커 파일 빌드**

./gradlew clean build -x test

**도커 허브에 이미지를 올리려면 login 필수**

docker login -u [username]

**도커 이미지 생성**

docker build -t [계정명]/[이미지명] .

**도커 이미지 목록**

docker images

**도커 허브에 이미지 푸시**

docker push [계정명]/[이미지명]

ec2에서 도커 허브에 올라온 이미지 pull docker pull [계정명]/[이미지명]

ec2에서 이미지 실행 sudo docker run -d -p 8081:8081 yskkjdh553/boot-prac

**-도커 이미지 배포(리액트)**

# 이름이 Dockerfile일 때는 -f 이름 부분 생략 가능

docker build -f Dockerfile -t yskkjdh553/react-prac ./

# 도커 허브에 이미지 푸시

docker push yskkjdh553/react-prac

# EC2에 접속해서 도커이미지 pull

docker pull yskkjdh553/react-prac

# Docker Image 컨테이너화

sudo docker run -d -p 3000:3000 yskkjdh553/react-prac

**4. 서비스 이용 방법**

1) 카카오 로그인  
준비 : 카카오 계정

2) 네이버 로그인

준비 : 네이버 계정