|  |
| --- |
|  |



|  |
| --- |
| **CRESCENDO** |
| **포팅 메뉴얼** |
|  |
|  |
|  |
|  |
|  |



목차

[I. 개요 2](#_Toc96072459)

[1. 프로젝트 개요 2](#_Toc96072460)

[2. 프로젝트 사용 도구 2](#_Toc96072461)

[3. 개발환경 2](#_Toc96072462)

[4. 외부 서비스 2](#_Toc96072463)

[5. GitIgnore 처리한 핵심 키들 3](#_Toc96072464)

[II. 빌드 3](#_Toc96072465)

[1. 환경변수 형태 3](#_Toc96072466)

[2. 빌드하기 5](#_Toc96072467)

[3. 배포하기 5](#_Toc96072468)

# 개요

## 프로젝트 개요

기존의 아이돌 덕질 커뮤니티는 폐쇄적인 면이 있고, 아는 사람들만 정보공유를 하기 때문에 정보습득에 있어서 제한적이다.이러한 점을 타파하고자 누구나 쉽게 접근하고 커뮤니티 컨텐츠를 참여함으로서 덕질에 흥미를 유발할 수 있습니다.

## 프로젝트 사용 도구

이슈 관리 : JIRA

형상 관리 : Gitlab

커뮤니케이션 : Notion, Mattermost

디자인 : Figma

UCC : 모바비

CI/CD : Jenkins

## 개발환경

VS Code

IntelliJ

React

Java : 17.0.11

Spring Boot: 3.2.8

Node.js : 14.16.0

SERVER : AWS EC2 Ubuntu 20.04.6 LTS

DB : MySQL

## 외부 서비스

MediaPipe

## Gitignore 처리한 핵심 키들

Spring Boot : application.yml

(\src\main\resources, 또는 classPath에 위치)

Flask : .env

# 빌드

## 환경변수 형태

.env.local :

// FIREBASE RTDB 프론트 키

VUE\_APP\_FIREBASE\_APIKEY=

VUE\_APP\_FIREBASE\_AUTHDOMAIN=

VUE\_APP\_FIREBASE\_DATABASEURL=

VUE\_APP\_FIREBASE\_PROJECTID=

VUE\_APP\_FIREBASE\_STORAGEBUCKET=

VUE\_APP\_FIREBASE\_MESSAGING\_SENDERID=

VUE\_APP\_FIREBASE\_APPID=

VUE\_APP\_FIREBASE\_MEASUREMENTID=

// YOUTUBE DATA API KEY

VUE\_APP\_YOUTUBE\_API\_KEY=

// 카카오 공유 서비스 키

VUE\_APP\_KAKAO\_API\_KEY=

.env

DB\_USER=DB호스트 유저 정보

DB\_PASS=DB 비밀번호

DB\_URL= DB 주소

.application.yml

server:

port: 포트번호

spring:

servlet:

multipart:

max-file-size: 25MB

max-request-size: 200MB

jpa:

open-in-view: false

properties:

hibernate:

default\_batch\_fetch\_size: 50

show\_sql: true

datasource:

driver-class-name: com.mysql.cj.jdbc.Driver

url: DB URL

username: 유저네임

password: DB비밀번호

sql:

init:

mode: always

data:

web:

pageable:

max-page-size: 2000

default-page-size: 10

mvc:

static-path-pattern: /files/\*\*

web:

resources:

static-locations: file:/home/ubuntu/files

jwt:

salt: jwt salt key

access-token:

expiretime: 3600000

refresh-token:

expiretime: 2592000000

mail:

host: smtp.naver.com

port: 587

id: 네이버 ID

pw: 네이버 비밀번호

image:

upload:

dir: /home/ubuntu/files

ai:

url: https://www.crescendo.o-r.kr/ai

## 빌드하기

1) Front

npm i

npm run build

2) Back-spring

Gradle 실행

Bootjar 실행

3) Back-flask

pip install -r requirements.txt

서버 구동 위치에서 python app.py

## 배포하기

Nginx 설정

server {

server\_name www.crescendo.o-r.kr;

client\_max\_body\_size 200M;

client\_body\_timeout 60s;

location / {

proxy\_pass http://localhost:3000;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

location /ai/ {

proxy\_pass http://localhost:5000/;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

location /server/ {

proxy\_pass http://localhost:8000/;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_buffering off;

proxy\_cache off;

proxy\_set\_header Connection '';

chunked\_transfer\_encoding off;

}

listen 443 ssl; # managed by Certbot

ssl\_certificate /etc/letsencrypt/live/www.crescendo.o-r.kr/fullchain.pem; # managed by Ce> ssl\_certificate\_key /etc/letsencrypt/live/www.crescendo.o-r.kr/privkey.pem; # managed by >

include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}

server {

if ($host = www.crescendo.o-r.kr) {

return 301 https://$host$request\_uri;

} # managed by Certbot

server\_name www.crescendo.o-r.kr;

listen 80;

return 404; # managed by Certbot

}

이후 sudo service nginx start