**포팅 매뉴얼**

목차

[개요 2](#_Toc147402684)

[프로젝트 개요 2](#_Toc147402685)

[사용한 기술 및 버전 2](#_Toc147402686)

[프론트 엔드 2](#_Toc147402687)

[백엔드 2](#_Toc147402688)

[외부 API 3](#_Toc147402689)

[빌드 3](#_Toc147402690)

[Spring boot의 application.yml 3](#_Toc147402691)

[빌드 하기 및 배포하기 5](#_Toc147402692)

[프론트 5](#_Toc147402693)

[백엔드 – spring boot 5](#_Toc147402694)

[Nginx 5](#_Toc147402696)

# 개요

## 프로젝트 개요

TV 예능프로그램에서 자주 나오는 노래 빨리 맞히기 게임을 더 많은 사람들과 웹 사이트에서 즐길 수 있도록 만든 웹 게임입니다.

저희 musiq는 로그인 없이 즐길 수 있는 게스트 모드, 로그인 후 혼자 즐길 수 있는 싱글모드 다른 사람들과 함께 즐길 수 있는 멀티 모드로 구성되어 있습니다.

## 사용한 기술 및 버전

### 프론트 엔드

* Node.js 18.16.1
* Npm 9.5.1
* React 18.2.0
* TypeScript 4.9.5
* Recoil 0.7.7
* Recoil-persist 5.1.0
* React query 3.39.3
* styled-components 6.1.0
* Framer Motion 10.15.0
* Prettier 3.0.0
* Axios 1.4.0
* React-router-dom 6.14.2
* Stompjs 7.0.0
* Eslint 8.45.0
* Prettier 3.0.0

### 백엔드

* Git 2.41.0.3
* IntelliJ 2021.2.4
* VSCode 1.80.1
* JDK 17.0.8.1(Amazon Coretto)
* Spring boot 3.1.3
* Docker Desktop 4.21.1
* MariaDB 11.0.2
* Gradle 8.1.1
* Redis 7.0.12
* Jenkins 2.429
* Nginx 1.18.0
* Amazon Ec2
* Spring-websocket 6.0.13
* Jpa Hibernate 8.0.1

## 외부 API

Kakao API : 카카오톡 공유기능을 구현하기 위해 사용.

# 빌드

## Spring boot의 application.yml

server:

servlet:

context-path: {어플리케이션의 Context-Path}

spring:

profiles:

active: dev

## Spring boot의 application-dev.yml

spring:

#mariadb

datasource:

url: jdbc:mariadb://{DB주소}/{DB이름}?useUnicode=true&characterEncoding=utf-8

driver-class-name: org.mariadb.jdbc.Driver

username: {DB 유저 이름}

password: {DB 비밀번호}

jpa:

open-in-view:

generate-ddl:

show-sql:

hibernate:

ddl-auto:

data:

redis:

host: {redis주소}

port: {redis 포트 번호}

password: {redis 비밀번호}

jwt:

secret-key: {JWT 시크릿 키}

## 빌드 하기 및 배포하기

### 프론트

* npm install && CI=’false’
* npm run build

### 백엔드 – spring boot

* gradlew 실행

## Nginx

server {

server\_name musiq.site;

root /jenkins/workspace/frontend\_asset/build;

index index.html index.htm;

location / {

try\_files $uri /index.html;

}

location /api {

proxy\_pass http://localhost:8080/api;

}

listen [::]:443 ssl ipv6only=on; # managed by Certbot

listen 443 ssl; # managed by Certbot

ssl\_certificate /etc/letsencrypt/live/musiq.site/fullchain.pem; # managed by Certbot

ssl\_certificate\_key /etc/letsencrypt/live/musiq.site/privkey.pem; # managed by Certbot

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

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

ssl\_protocols TLSv1.2 TLSv1.3;

location /api/game-websocket {

proxy\_pass http://localhost:8080/api/game-websocket;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "upgrade";

# proxy\_redirect off;

# 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-Host $server\_name;

}

}

#http 요청이 들어오면 https 로 바꿔서 요청보내는 곳

server {

if ($host = musiq.site) {

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

} # managed by Certbot

listen 80;

listen [::]:80;

server\_name musiq.site;

return 404; # managed by Certbot

}

}

이후, sudo service nginx restart