**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.properties 3](#_Toc147402691)

[빌드 하기 및 배포하기 5](#_Toc147402692)

[프론트 5](#_Toc147402693)

[백엔드 – spring boot 5](#_Toc147402694)

[백엔드 – fastAPI 5](#_Toc147402695)

[Nginx 5](#_Toc147402696)

# 개요

## 프로젝트 개요

주식을 할 때 혹은 취업 준비를 위해서 기업 분석을 할 때, 해당 기업의 재무제표는 아주 중요한 지표 입니다. 하지만, 재무제표를 학습 하기 위한 자료들이 별로 없고 너무 어려워서 쉽게 학습을 할 수가 없습니다. 그래서 저희는 이러한 불편함을 겪고, 퀴즈를 통해서 재미있고 쉽게 재무제표를 학습하는 서비스를 만들려고 했습니다.

## 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
* Emotion 11.11.1
* Emotion/styled 11.11.0
* Framer Motion 10.15.0
* Prettier 3.0.0
* Axios 1.4.0
* React-router-dom 6.14.2
* Storybook ^7.4.1

### 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
* Python 3.9.13
* fastAPI 0.103.2

## 외부 API

Google Bard API : 뉴스 크롤링한 정보를 요약하기 위해서 사용, 현재 사용하는 버전의 이슈 때문에 0.1.34를 사용

# 빌드

## Spring boot의 application.properties

#context-path

server.servlet.context-path=

#applcation port ??

server.port=사용자가 사용할 포트

#mariadb

spring.datasource.driver-class-name=org.mariadb.jdbc.Driver

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

spring.datasource.username={DB 유저 이름}

spring.datasource.password={DB 비밀번호}

#jpa

spring.jpa.hibernate.ddl-auto=

spring.jpa.properties.hibernate.format\_sql=

spring.jpa.show-sql=

#redis

spring.redis.host={redis주소}

spring.redis.port=

logging.level.com.ssafy.jazz\_backend.domain.websocket.controller=INFO

logging.level.com.ssafy.jazz\_backend.domain.websocket.EventListener=INFO

#API\_KEY

dictionary.serviceKey={금융결제원 API 키}

#Kafka

spring.kafka.bootstrap-servers={kafka broker주소}

spring.kafka.consumer.key-deserializer=org.apache.kafka.common.serialization.StringDeserializer

spring.kafka.consumer.value-deserializer=org.apache.kafka.common.serialization.StringDeserializer

spring.jackson.serialization.fail-on-empty-beans=

## 빌드 하기 및 배포하기

### 프론트

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

### 백엔드 – spring boot

* gradlew 실행

### 백엔드 – fastAPI

* pip install -r requirements.txt
* uvicorn main:app –reload

## Nginx

server {

root /home/ubuntu/jazz\_frontend/dist/build;

server\_name j9a708.p.ssafy.io;

location / {

# First attempt to serve request as file, then

# as directory, then fall back to displaying a 404.

index index.html;

try\_files $uri $uri/ /index.html;

}

location /api/ {

proxy\_pass http://localhost:8081;

}

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

listen 443 ssl; # managed by Certbot

ssl\_certificate /etc/letsencrypt/live/j9a708.p.ssafy.io/fullchain.pem; # managed by Certbot

ssl\_certificate\_key /etc/letsencrypt/live/j9a708.p.ssafy.io/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

}

#http 요청이 들어오면 https 로 바꿔서 요청보내는 곳

server {

if ($host = j9a708.p.ssafy.io) {

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

} # managed by Certbot

listen 80 default\_server;

listen [::]:80 default\_server;

server\_name j9a708.p.ssafy.io;

return 404; # managed by Certbot

}

이후, sudo service nginx restart