



포팅 메뉴얼

☰ 태그	관리
📅 마감일	@2024년 5월 20일
📉 진행 상황	In Progress

Frontend

- VS Code 1.86.2

```
### 프로젝트 정보
- **이름**: frontend
- **버전**: 0.0.0
- **모듈 유형**: ES 모듈 (type: module)

### 스크립트
- **개발 서버 실행**: `npm run dev`
- **빌드**: `npm run build`
- **린트**: `npm run lint`
- **프리뷰**: `npm run preview`

### 주요 의존성
- **스타일링 및 UI**:
  - Emotion 라이브러리 (`@emotion/css`, `@emotion/react`, `@emotion/styled`)
  - Radix UI 라이브러리 (`@radix-ui/react-*`)
  - Tailwind CSS (`tailwindcss`, `tailwindcss-animate`, `tailwind-merge`)
- **폼 관리**:
  - React Hook Form (`@hookform/resolvers`, `react-hook-form`)
- **데이터 시각화**:
  - Nivo 라이브러리 (`@nivo/bar`, `@nivo/core`, `@nivo/pie`)
```

- ****에디터 확장****:
 - TipTap 에디터 라이브러리 (`@tiptap-pro/extension-*`, `@tiptap/extension-*`, `@tiptap/react`, `@tiptap/starter-kit`)
- ****라우팅****:
 - React Router DOM (`react-router-dom`)
- ****기타****:
 - HTTP 클라이언트 (`axios`)
 - 날짜 처리 (`date-fns`)
 - 코드 하이라이팅 (`highlight.js`, `lowligh`)
 - 코드 포매터 (`prettier`, `prettier-plugin-tailwindcss`)
 - 타입 체크 (`prop-types`)
 - React 라이브러리 (`react`, `react-dom`)
 - 데이터 패칭 (`react-query`)
 - 상태 관리 (`zustand`)

개발 의존성

- ****TypeScript****:
 - TypeScript (`typescript`)
 - ESLint 플러그인 및 파서 (`@typescript-eslint/eslint-plugin`, `@typescript-eslint/parser`)
- ****빌드 및 번들링****:
 - Vite 빌드 도구 및 플러그인 (`vite`, `@vitejs/plugin-react`, `vite-plugin-svg`)
- ****ESLint****:
 - ESLint 및 관련 플러그인 (`eslint`, `eslint-config-airbnb`, `eslint-config-prettier`, `eslint-plugin-prettier`, `eslint-plugin-react-hooks`, `eslint-plugin-react-refresh`)
- ****기타****:
 - TypeScript 타입 정의 (`@types/node`, `@types/react`, `@types/react-dom`, `@types/react-lottie`, `@types/react-table`)
 - CSS 후처리기 (`postcss`, `autoprefixer`)

Backend

- IntelliJ IDEA 2024.1

- JAVA JDK 17
- Spring Boot 3.2.5
- Gradle 8.7
- JVM 17.0.10

Webserver

NGINX 1.25.5

after-each-Front-branch-build.sh (NGINX 설정 파일)

```
# /home/ubuntu/after-each-Front-branch-build.sh

#!/bin/bash

# Docker 컨테이너에 접속하여 다음 명령어들을 실행
docker exec -i pubble-nginx-frontend /bin/bash -c "
# Nginx 설정 파일 내용 대체
cat > /etc/nginx/conf.d/default.conf << 'EOF'
# HTTP server configuration
server {
    listen 80;
    listen [::]:80;
    server_name k10d109.p.ssafy.io;

    # HTTP to HTTPS redirect
    location / {
        return 301 https://\${host}\$request_uri;
    }
}

# HTTPS server configuration
server {
    listen 443 ssl;
    listen [::]:443 ssl;
    server_name k10d109.p.ssafy.io;
```

```

# SSL certificate storage
ssl_certificate /etc/letsencrypt/live
/k10d109.p.ssafy.io/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live
/k10d109.p.ssafy.io/privkey.pem;

# Static files serving web server
location / {
    root /usr/share/nginx/html;
    index index.html index.htm;
    try_files $uri /index.html;
}

# WebSocket configuration for STOMP
location /api/ws {
    proxy_pass http://k10d109.p.ssafy.io:10425;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    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;

    # Timeout adjustments
    proxy_read_timeout 86400; # 24h timeout
    proxy_connect_timeout 86400; # 24h timeout
    proxy_send_timeout 86400; # 24h timeout
}

# Reverse proxy for API requests
location /api {
    proxy_pass http://k10d109.p.ssafy.io:10425;
    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};
    }
}
EOF

# Nginx 설정 적용
nginx -s reload
"

```

환경변수

Frontend

```

# .env.production
VITE_API_URI=https://k10d109.p.ssafy.io/api/
VITE_SCREENSHOT_API=fX40brFpZvxM8w
VITE_TIPTAP_APP_ID=w9nqrw9o
VITE_TIPTAP_APP_SECRET=62b4a241943b178199d86a0918499
cb1c9b8c4c5eaeba4b26c33ef410f61e3fa
VITE_TIPTAP_API_SECRET=382fdab5606e973dd4ae73b61833e55
1e21ad38383756bbd6ec55f3eb2deb2ef
VITE_FIREBASE_VAPID_KEY=BKhDPAh7wlvs2zaucdkXq4_BgIfco9
KEqqPBmt-3kF05IojTMGizvdXEVsd5nS-WNrJGnZ7VsC4c_ri8U30yAyU
VITE_STOMP_BROKER_URL=k10d109.p.ssafy.io/api/ws

# .npmrc
@tiptap-pro:registry=https://registry.tiptap.dev/
//registry.tiptap.dev/:_authToken=xW1TizxyL76FwGtiJLQ/UOWfYN
RT6J8NfvkwjWWKSL1IJ7UAR1jLk2F3Z2oYg3gY

docker run -d --restart always -p 80:80 -p 443:443
-e TZ=Asia/Seoul -v /etc/letsencrypt/:/etc/letsencrypt/
--name pubble-nginx-frontend -u root ${dockerHubRepo}:latest"

```

Backend

```
JASYPT_PASSWORD=niveavaseline1016!  
FRONT_BASE=http://k10d109.p.ssafy.io:10425/api/  
FRONT_BASE_SECURE=https://k10d109.p.ssafy.io:10425/api/  
FRONT_LOCAL=http://localhost:5173  
FRONT_LOCAL_SECURE=https://localhost:5173  
BASE_URL=https://k10d109.p.ssafy.io  
hyojukimdev/pubble-backend:latest
```

배포 시 특이사항

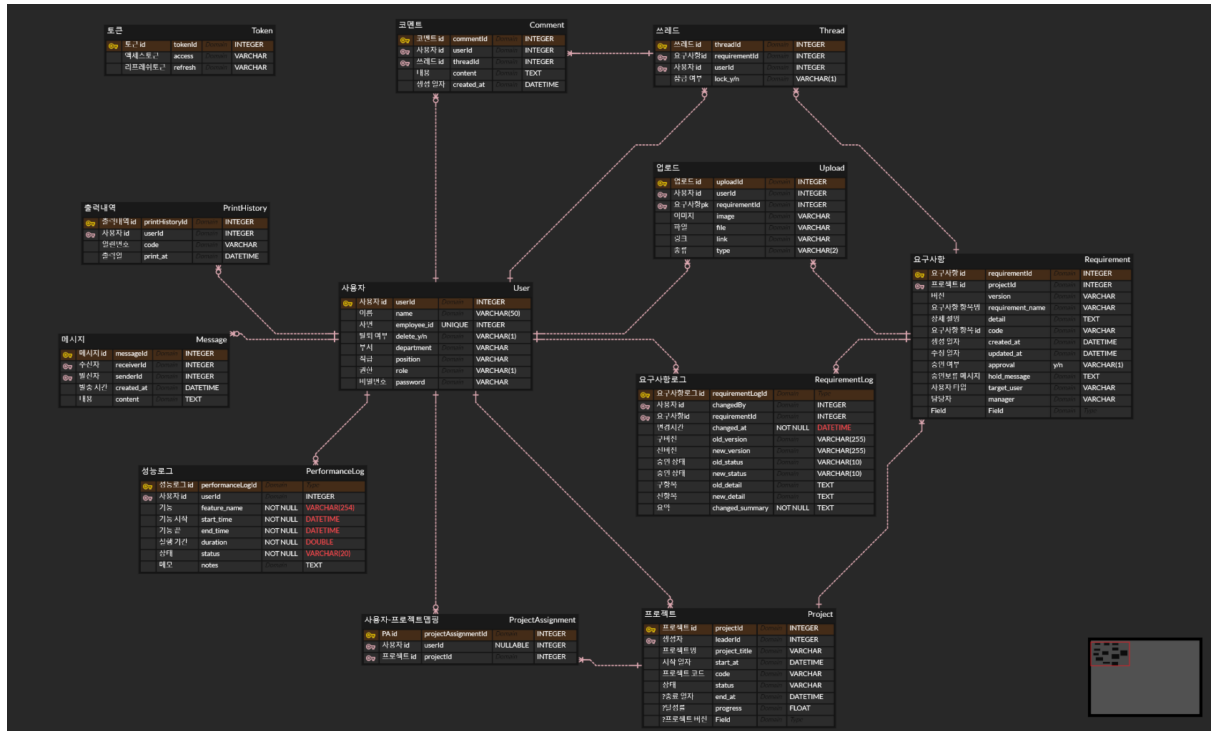
- Docker 사용
 - Frontend

```
docker run -d --restart always -p 80:80 -p 443:443  
-e TZ=Asia/Seoul -v /etc/letsencrypt/:/etc/letsencrypt/  
--name pubble-nginx-frontend -u root ${dockerHubRepo}:late
```

- Backend

```
docker run -d -p 10425:9091 --name pubble-backend  
-e JASYPT_PASSWORD=niveavaseline1016!  
-e FRONT_BASE=http://k10d109.p.ssafy.io:10425/api/  
-e FRONT_BASE_SECURE=https://k10d109.p.ssafy.io:10425/api/  
-e FRONT_LOCAL=http://localhost:5173 -e  
FRONT_LOCAL_SECURE=https://localhost:5173  
-e BASE_URL=https://k10d109.p.ssafy.io  
hyojukimdev/pubble-backend:latest
```

ERD



DB 접근 계정

DB 종류 : mySQL

- 외부 접근 port : 10420
- 계정명 : root
- 비밀번호 : potato1635!

테스트 계정

ID : SSAFY1005

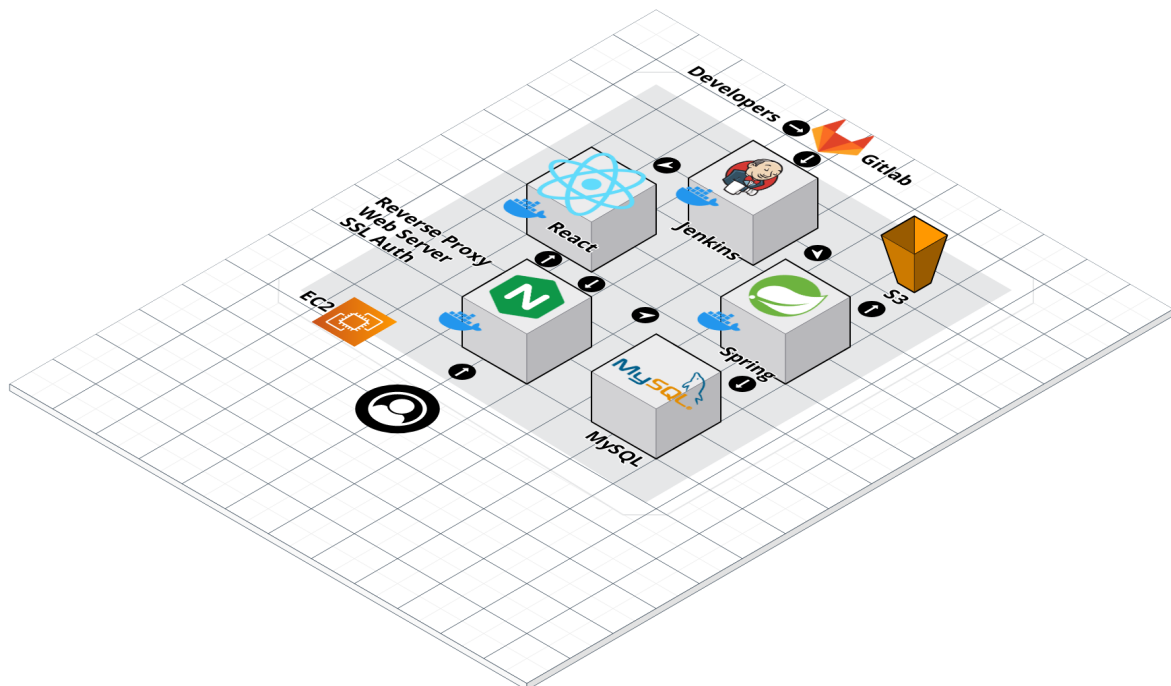
PW : 1234

프로젝트에서 사용하는 외부 서비스 정보

- Tiptap Cloud

- APP_ID= `w9nqrw9o`
- APP_SECRET= `62b4a241943b178199d86a0918499cb1c9b8c4c5ea`
`eba4b26c33ef410f61e3fa`
- API_SECRET= `382fdab5606e973dd4ae73b61833e551e21ad3838`
`3756bbd6ec55f3eb2deb2ef`
- 스크린샷
 - <https://screenshotone.com/>
 - SCREENSHOT_API = `fx40brFpZvxM8w`
- Firebase
 - VAPID_KEY= `BKhDPAh7wlvS2zaucdkXq4_BgIfco9KEqqPBmt-3kF05I`
`ojTMGizvdXEvsd5nS-WNrJGnZ7VsC4c_ri8U30yAyU`

Architecture



시연 시나리오

- 시연 시나리오 : 시연 순서에 따른 site 화면별, 실행별(클릭 위치 등) 상세 설명