



고민 한 접시

포팅 메뉴얼

목차

- I. 개요 2
 - 1. 프로젝트 개요 2
 - 2. 프로젝트 사용 도구..... 2
 - 3. 개발환경 2
 - 4. 외부 서비스 3
 - 5. GITIGNORE 처리한 핵심 키들 3
- II. 빌드 3
 - 1. 환경변수 형태 3
 - 2. 빌드하기 4
 - 3. 배포하기 5

I. 개요

1. 프로젝트 개요

“오늘 올려드릴 초밥은 ‘고민 한 접시’입니다. 차분하게 숙성된 고민을 감싸 안은 초밥으로, 깊은 생각의 결이 느껴지는 것이 특징입니다. 이 고민을 한입 베어 물면, 바쁜 하루 속에서 미처 털어놓지 못한 이야기들이 자연스럽게 퍼져 나갑니다. 조심스럽게 음미하시고, 떠오르는 생각을 함께 엿어 보셔도 좋겠습니다.”

2. 프로젝트 사용 도구

이슈 관리 : JIRA
형상 관리 : Gitlab
커뮤니케이션 : Notion, Mattermost
디자인 : Figma
UCC : 모바비, 애프터이펙트, 프리미어
CI/CD : Jenkins

3. 개발환경

	Skill	Version
Back	Java	17
	SpringBoot	3.4.1
	IntelliJ	21.0.5+8-b631.30 amd64
Front	VS Code	1.97.2
	Node.js	22.12.0
	NPM	10.9.0
Data	MariaDB(azure)	
	Redis	7.4.2
Server	AWS EC2	Ubuntu 22.04.5 LTS

EC2 포트 번호

skill	Port
Back	8080
MariaDB	3306
Redis	6379
Nginx	80/443
Jenkins	8081

4. 외부 서비스

Google OAuth : application-secret.yml 에 해당 내용 있음

KakaoAPI : application-secret.yml 에 해당 내용 있음

chatGPT : application-secret.yml 에 해당 내용 있음

5. Gitignore 처리한 핵심 키들

Vite : .env (최상단 위치)

Spring : application-secret.yml

(\src\main\resources, 또는 classPath 에 위치)

II. 빌드

1. 환경변수 형태

.env :

```
// API 요청 BASE URL
VITE_API_BASE_URL=
// 카카오 로그인 클라이언트 ID
VITE_KAKAO_CLIENT_ID=
// 카카오 공유 서비스 키
VITE_KAKAO_JAVASCRIPT_ID=
// 구글 로그인 클라이언트 ID
VITE_GOOGLE_CLIENT_ID=
```

application-secret.yml :

```
# 도메인
app:
  domain: 도메인

# DB
spring:
  datasource:
    url: ${DB_URL}
    username: ${DB_USERNAME}
    password: ${DB_PASSWORD}
    driver-class-name: org.mariadb.jdbc.Driver
  data:
    redis:
      password: 비밀번호
      port: 포트번호

# Security OAuth
oauth:
  kakao:
    client-id: 클라이언트 아이디
  google:
    client-id: 클라이언트 아이디
    client-secret: 비밀번호
#   redirect-uri 는 각 profile 에 설정

# JWT 시크릿 키
jwt:
  secret: 시크릿 키

# chatgpt 키
api:
  chatgpt:
    key: chatgpt API 키
```

2. 빌드하기

1) Front

```
npm i
npm run build
```

2) Back-spring

```
Gradle 실행
Bootjar 실행
```

3. 배포하기

Nginx 설정

- 해당 위치에 설정 파일 생성

```
sudo vi /etc/nginx/sites-available/project.conf
```

- 심볼릭 링크 생성

```
sudo ln -s /etc/nginx/sites-available/project.conf /etc/nginx/sites-enabled/project.conf
```

- /etc/nginx/conf.d/service-url.inc

```
set $service_url http://127.0.0.1:8080;
```

project.conf

```
server {
    if ($host = gomin.my) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    if ($host = www.gomin.my) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80;
    server_name www.gomin.my gomin.my;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name www.gomin.my gomin.my;
    ssl_certificate /etc/letsencrypt/live/gomin.my/fullchain.pem; #
managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/gomin.my/privkey.pem; #
managed by Certbot

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers HIGH:!aNULL:!MD5;

    location / {
        root /home/ubuntu/dist;
        index index.html index.htm;
        try_files $uri $uri/ /index.html;
    }

    include /etc/nginx/conf.d/service-url.inc;

    # API 요청을 백엔드 서버(포트 8080)로 프록시
    location /api {
```

```
    proxy_pass $service_url;
    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;
}

# SSE 엔드포인트
location /api/sse/subscribe {
    proxy_pass $service_url;
    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 '';
    proxy_http_version 1.1;
    chunked_transfer_encoding on;

    # timeout 설정 추가
    proxy_read_timeout 185s;
    proxy_send_timeout 185s;
    proxy_connect_timeout 5s;
}
}
```

이후 `sudo systemctl restart nginx`