오픈티콘

포팅 매뉴얼

B204팀

박찬수, 김용원, 강도원, 양종원, 김용원, 박병준

목차

1. 개요
   1. 프로젝트 개요
   2. 프로젝트 사용도구
   3. 개발환경
   4. 외부 서비스
   5. GitIgnore 처리한 핵심 키
2. 빌드
   1. 환경변수 형태
   2. 빌드하기
3. 배포
   1. 배포하기
   2. 서비스 이용 방법
4. 개요
   1. 프로젝트 개요

친한 사람들과 메신저를 통해 소통하다보면 최근 인터넷에 유행하는 밈(meme)을 공유할 때가 있다. 하지만 항상 모든 밈을 휴대전화에 저장하고 전송할 준비를 하기는 쉽지않다. 또 밈이 이모티콘화 되어 공유되는 현재, 기성 이모티콘 서비스는 다소 긴 심사기간을 가지므로 유행에 발빠르게 대응하기 어렵다. 따라서 각 사용자가 자유롭게 이모티콘을 업로드하고 다른 사용자와 간편하게 공유할 수 있는 서비스 ‘오픈티콘’을 제안한다. ‘오픈티콘’은 사용자가 제작한 이모티콘을 간단한 심사와 커뮤니티 자정작용을 통해 빠르게 출시할 수 있도록 하고, 생성형 인공지능을 활용해 누구나 작가가 될 수 있는 기회를 제공한다. 또 안드로이드 애플리케이션을 통해 구매한 이모티콘을 자주 사용하는 메신저에서 플로팅 메뉴를 통해 간단하게 사용할 수 있어 사용자의 경험을 기존 서비스 대비 한단계 진보시켰다.

* 1. 프로젝트 사용도구
     1. 이슈 관리 : JIRA
     2. 소프트웨어 형상 관리 : Gitlab
     3. DB 형상 관리 : Flyway
     4. 커뮤니케이션 : Notion, Mattermost, Discord
     5. 디자인 : Figma
     6. UCC : 모바비, Vrew.ai
     7. CI/CD : Jenkins, Docker,
  2. 개발환경
     1. VS Code 1.93.1, Cursor IDE 0.42.5
     2. IntelliJ : 2024.2.4 RC amd64
     3. Android Studio 24.2.1 amd64
     4. JVM : eclipse-temurin:17 (빌드 환경: gradle:8.10.2-jdk17)
     5. SERVER : AWS EC2 Ubuntu 20.04.3 LTS
     6. DB : PostgreSQL 17 (Ubuntu 17.0-1.pgdg20.04+1)
     7. Docker version 27.3.1, build ce12230
  3. 외부 서비스

1. Google Cloud API 키: 개인별로 배포한 detectsafesearch json 파일에 해당 내용있음. (과금발생 가능성 존재하므로 취급에 주의할 것)
2. 네이버, 구글, 카카오 소셜 로그인 인증키: 개인별로 배포한 env파일 및 Jenkins내 파이프라인 스크립트에 해당 내용 있음
3. 아임포트 인증키: 개인별로 배포한 env파일 및 Jenkins내 파이프라인 스크립트에 해당 내용 있음
   1. GitIgnore 처리한 핵심 키

Vue : .env (최상단 위치)

Spring : .env detectsafesearch-df3e757877a2.json

1. 빌드
   1. 환경변수 형태

//프로젝트 이름

ENV\_PROJECT\_NAME="openticon-apitest"

// 프로젝트 활성화 프로필

ENV\_SPRING\_PROFILES\_ACTIVE="dev"

// 데이터베이스 접속정보

ENV\_DEV\_NAME="ssafy"

ENV\_DEV\_PASSWORD=" "

// 카카오 소셜 로그인

ENV\_KAKAO\_OAUTH\_CLIENT\_ID=""

ENV\_KAKAO\_OAUTH\_CLIENT\_SECRET=" "

// 네이버 소셜 로그인

ENV\_NAVER\_OAUTH\_CLIENT\_ID=" "

ENV\_NAVER\_OAUTH\_CLIENT\_SECRET=""

// 구글 소셜 로그인

ENV\_GOOGLE\_OAUTH\_CLIENT\_ID=" "

ENV\_GOOGLE\_OAUTH\_CLIENT\_SECRET=" "

// JWT 시크릿 키

ENV\_JWT\_SECRET\_KEY=" "

// 아임포트 API 키

ENV\_IAMPORT\_API\_KEY=""

ENV\_IAMPORT\_API\_SECRET=" "

// 배포 서버 주소

ENV\_DEV\_BASE\_URL="https:// "

// 사용자가 접속할 이미지 서버 주소

ENV\_DEV\_IMAGE\_SERVER\_URL="https:// "

// 백엔드 서버 노출 포트

ENV\_SPRING\_EXPOSE\_PORT="8080"

// 이미지 서버 노출 포트

ENV\_IMAGE\_SERVER\_EXPOSE\_PORT="8070"

// 데이터베이스 노출 포트

ENV\_POSTGRES\_EXPOSE\_PORT="5432"

// 구글 세이프 서치 Json 경로

ENV\_GOOGLE\_APPLICATION\_CREDENTIALS="/app/detectsafesearch-df3e757877a2.json"

// GPU 서버 주소

ENV\_DEV\_GPU\_URL="http:// "

* 1. 빌드하기

1) Front

npm install

npm run build

2) Back-spring

Gradle 실행

Build task 실행

도커 컴포즈 파일로 프론트엔드, 백엔드 각각 실행 가능. 각 프로젝트의 env파일을 반영하여 적용.

1. 배포
   1. 배포하기

Caddy 설정

code.openticon.store {

reverse\_proxy 172.19.0.1:13373

}

apitest.openticon.store {

handle /api/\* {

reverse\_proxy openticon-apitest-spring-1:8080

request\_body {

max\_size 100000000MB

}

}

handle /static/\* {

reverse\_proxy openticon-apitest-express-1:8070

}

handle /\* {

reverse\_proxy frontend-frontend-1:80

}

}

share.openticon.store {

handle /\* {

reverse\_proxy frontend-frontend-1:81

}

}

portainer.openticon.store {

reverse\_proxy portainer:9000

}

coder.openticon.store {

reverse\_proxy coder:7080

}

* 1. 외부 서비스 이용 방법

가) Google Cloud API 설정

API 키 생성

GCP 웹사이트에서 계정 생성 후 로그인

API 키를 생성하고, 보안을 위해 env 파일에 추가 (gitignore 확인)

나) API 요청 코드 작성

이미지 생성 API 호출

사용 중인 언어에 맞는 GCP 클라이언트 라이브러리를 설치

API 키를 설정하고, 이미지 생성을 위한 코드를 작성

다) 웹 프로젝트와 통합

프론트엔드와 백엔드 통합

프론트엔드에서 사용자 입력을 받아 API 요청을 백엔드에 전송

백엔드에서 GCP API를 호출하여 사용자 이미지 검사

프론트엔드에서 이미지를 표시

라) 비용 및 사용량 관리

API 사용량 모니터링

GCP 대시보드에서 API 사용량과 비용을 정기적으로 확인

예산을 초과하지 않도록 모니터링하고, 필요시 호출 최적화 적용