|  |
| --- |
|  |



|  |
| --- |
| **TUPLI ~Youtube Playlist~** |
| **포팅 메뉴얼** |
|  |
|  |
|  |
|  |
|  |



목차

[I. 개요 2](#_Toc96051433)

[1. 프로젝트 개요 2](#_Toc96051434)

[2. 프로젝트 사용 도구 2](#_Toc96051435)

[3. 개발환경 2](#_Toc96051436)

[4. 외부 서비스 2](#_Toc96051437)

[5. Gitgnore 처리한 핵심 키들 3](#_Toc96051438)

[II. 빌드 3](#_Toc96051439)

[1. 환경변수 형태 3](#_Toc96051440)

# 개요

## 프로젝트 개요

콘텐츠의 범람이 일어나는 요즘 시대에서 여러분들은 유튜브를 보며 '이 영상, 나만보기 아깝다!' 또는, '나와 비슷한 취향을 가진 사람은 어떤 채널을 구독중일까?' 라는 생각을 해보신 적 있나요?

TUPLI는 그러한 니즈를 충족시켜주기 위해 탄생한 유튜브 콘텐츠 기반 SNS입니다. TUPLI와 함께라면, 유튜브 컨텐츠 공유를 손쉽게! 할 수 있습니다.

## 프로젝트 사용 도구

이슈 관리 : JIRA

형상 관리 : Gitlab

커뮤니케이션 : Notion, Slack, Mattermost

디자인 : Figma

UCC : 모바비, 애프터이펙트, 프리미어

CI/CD : Jenkins

## 개발환경

VS Code : 1.64.2,

IntelliJ : 11.0.13+7-b1751.21 amd64

JVM : 16.0.1 (스프링은 11로 빌드)

Node.js : 14.16.0

SERVER : AWS EC2 Ubuntu 20.04.3 LTS

DB : MariaDB (azure)

## 외부 서비스

Google OAuth : application.yml에 해당 내용 있음

Google Cloud Storage : serviceAccountKey.json에 해당 내용 있음  
(과금이 발생할 수 있는 키입니다. 취급 주의)

Firebase Realtime DB (back) : serviceAccountKey.json에 해당 내용 있음  
(과금이 발생할 수 있는 키입니다. 취급 주의)

Redis cloud : application.yml에 해당 내용 있음

KakaoPayAPI : application.yml에 해당 내용 있음

Firebase Realtime DB (front) :

## Gitgnore 처리한 핵심 키들

Vue : .env.loval (최상단 위치)

Spring : application.yml, serviceAccountKey.json

(\src\main\resources, 또는 classPath에 위치)

Flask : .env (최상단 위치)

# 빌드

## 환경변수 형태

.env.local :

// FIREBASE RTDB 프론트 키

VUE\_APP\_FIREBASE\_APIKEY=

VUE\_APP\_FIREBASE\_AUTHDOMAIN=

VUE\_APP\_FIREBASE\_DATABASEURL=

VUE\_APP\_FIREBASE\_PROJECTID=

VUE\_APP\_FIREBASE\_STORAGEBUCKET=

VUE\_APP\_FIREBASE\_MESSAGING\_SENDERID=

VUE\_APP\_FIREBASE\_APPID=

VUE\_APP\_FIREBASE\_MEASUREMENTID=

// YOUTUBE DATA API KEY

VUE\_APP\_YOUTUBE\_API\_KEY=

// 카카오 공유 서비스 키

VUE\_APP\_KAKAO\_API\_KEY=

.env

DB\_USER=DB호스트 유저 정보

DB\_PASS=DB 비밀번호

DB\_URL= DB 주소

.application.yml

# REDIS

redis:

database: 0

host: 호스트 주소

password: 비밀 번호

port: 포트 번호

# 마리아 DB(배포)

datasource:

url: 마리아 DB 주소

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

username: 유저 이름

password: 유저 비밀번호

# Security OAuth

security:

oauth2.client:

registration:

google:

clientId: 클라이언트 아이디

clientSecret: 비밀번호

scope:

- email

- profile

# JWT 시크릿 키

jwt:

secret: 시크릿 키

# 카카오 ADMIN 키

key:

kakao:

admin: 카카오 페이 API 키

.serviceAccountKey

Google Cloud Platform 유료 회원이 지급 받을 수 있는 프로젝트 전권한 루트 key

## 빌드하기

1) Front

npm i

npm run build

2) Back-spring

Gradle 실행

Bootjar 실행

3) Back-flask

pip install -r requirements.txt

서버 구동 위치에서 python app.py

## 배포하기

Nginx 설정

upstream jenkins {

server 127.0.0.1:4000;

keepalive 32;

}

map $http\_upgrade $connection\_upgrade {

default upgrade;

'' close;

}

server {

listen 80 default\_server;

listen [::]:80 default\_server;

server\_name tupli.kr;

return 301 https://$server\_name$request\_uri;

}

server {

listen 443 ssl;

listen [::]:443 ssl;

server\_name tupli.kr;

ssl\_certificate /etc/letsencrypt/live/tupli.kr/fullchain.pem;

ssl\_certificate\_key /etc/letsencrypt/live/tupli.kr/privkey.pem;

root /var/www/html/dist;

index index.html;

location / {

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

}

location /api/v1 {

proxy\_pass http://localhost:8080;

proxy\_redirect off;

charset utf-8;

rewrite /api/v1/(.\*) /$1 break;

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\_set\_header X-NginX-Proxy true;

}

location /api/v1/ws-stomp {

proxy\_pass http://localhost:8080;

rewrite /api/v1/(.\*) /$1 break;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "Upgrade";

proxy\_set\_header Host $host;

}

location /flask {

proxy\_pass http://localhost:5000;

proxy\_redirect off;

charset utf-8;

rewrite /flask/(.\*) /$1 break;

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\_set\_header X-NginX-Proxy true;

}

}

이후 sudo service nginx start

## 외부 서비스 이용 방법

### 카카오 페이

준비 ; 카카오페이 API 등록

- 내 애플리케이션 → 애플리케이션 추가하기

- 들어갈 경우 Admin키가 있음 → 보안을 위해 yml 파일에 추가(gitgnore 했는지 확인!)

- 플랫폼 등록 ⇒ 웹 (적는것은 프론트!)

### Firebase 실시간 알림

1. Firebase 콘솔 프로젝트 만들기 (GCP에 프로젝트 있을 경우 그거 선택)

2. 설정 → 프로젝트 설정 → 서비스 계정 → 새 비공개 키 생성

3. realtimedatabase 생성 후 주소 확인

### Google Cloud Platform

1. Google Cloud Platform 접속 후 새 프로젝트 만들기

→ 좌측 햄버거 메뉴에서 저장소 Cloud Storage 클릭

2. 브라우저로 이동될텐데 버킷 만들기 선택 후 버킷 이름 지정하고 만들기

일단은 tupli-profile로 생성 (이 버킷 이름은 나중에 spring에서 저장 위치로 사용됨)

3. springboot 접속용 access key를 만들어야 함.

좌측 햄버거 메뉴 다시 열고 IAM 및 관리자 > 서비스 계정 으로 이동후

서비스 계정 만들기 클릭

4. 서비스 계정이름 대충 만들고 (tupli-profile-access),

권한 부여에 저장소 관리자, 저장소 개체 관리자 부여

5. 서비스 계정 만들고 나올 경우, 속성 등으로 키로 이동

키 추가 선택하여 json 다운로드