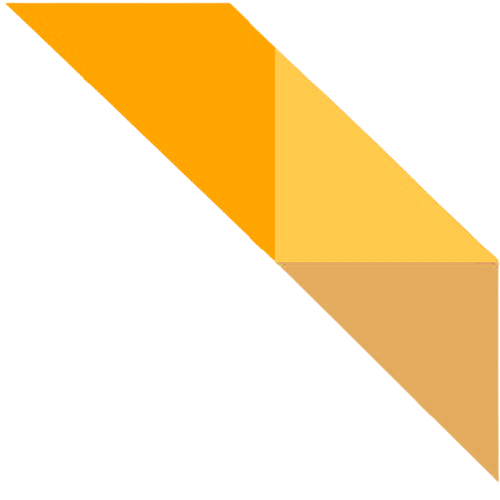
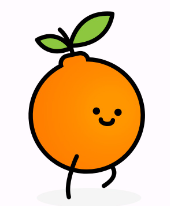
|  |
| --- |
|  |



|  |
| --- |
| **여행을 정산하다, 여정** |
| **포팅 메뉴얼** |
|  |
|  |
|  |
|  |
|  |



목차

[I. 개요 2](#_Toc96072459)

[1. 프로젝트 개요 2](#_Toc96072460)

[2. 프로젝트 사용 도구 2](#_Toc96072461)

[3. 개발환경 2](#_Toc96072462)

[4. 외부 서비스 2](#_Toc96072463)

[5. Gitgnore 처리한 핵심 키들 3](#_Toc96072464)

[II. 빌드 3](#_Toc96072465)

[1. 환경변수 형태 3](#_Toc96072466)

[2. 빌드하기 5](#_Toc96072467)

[3. 배포하기 5](#_Toc96072468)

[4. 서비스 이용 방법 7](#_Toc96072469)

[가) 카카오 로그인 7](#_Toc96072470)

[나) Firebase 실시간 알림 8](#_Toc96072471)

[다) S3 8](#_Toc96072472)

# 개요

## 프로젝트 개요

## 목표: 여행 그룹의 회원들이 여행 기간 동안 발생한 비용을 간편하게 정산하고 관리할 수 있는 자동 정산 서비스를 제공함으로써, 사용자들의 여행 경험을 향상시킨다.

친구들과 신나는 여행을 마치면 항상 기다리고 있는 귀찮은 일이 있습니다.

바로 정산입니다.

서로 얼마를 사용했고 누구에게 얼마를 요청하고 너무 복잡하지 않았나요?

여정에서 정산을 손쉽게 만들어 드립니다.

서로 복잡한 계산과 애매하게 남은 금액을 매번 결제한 사람이 부담하는 불편함도

주기도 받기도 애매한 금액으로 정산이 곤란한 경우도 여정에서 도와드립니다.

## 프로젝트 사용 도구

이슈 관리 : JIRA

형상 관리 : Gitlab

커뮤니케이션 : Notion, Mattermost

BE : SpringBoot3, Java17, JPA

FE : Flutter (14.16.0), Dart (3.3.1), DIO (5.4.1)

SERVER : AWS EC2 Ubuntu 20.04.3 LTS

DB : MariaDB (10.6.17)

## 외부 서비스

Kakao OAuth : 서브 모듈로 관리

Firebase FCM (BE) : yeojung\_firebase\_service\_key.json에 설정 내용 있음

Redis cloud : application.properties에 설정 내용 있음

Amazon S3 (BE) : application.properties에 설정 내용 있음

NAVER CLOVA OCR (BE) : application.properties에 설정 내용 있음

## Gitgnore 처리한 핵심 키들

Flutter : 서브모듈 사용

Spring : application.properties, yeojung\_firebase\_service\_key.json

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

# 빌드

## 환경변수 형태

application.properties

# REDIS

spring.data.redis.host=redis-10500.c74.us-east-1-4.ec2.cloud.redislabs.com

spring.data.redis.port=10500

spring.data.redis.password=qUvoSAD6uFSmsHxmWKPNorjIsO6x4ExM

#spring.data.redis.ssl.enabled=true

# MariaDB

spring.datasource.driverClassName=org.mariadb.jdbc.Driver

spring.datasource.url=jdbc:mariadb://j10c203.p.ssafy.io:3306/yeojung

spring.datasource.username=root

spring.datasource.password=yeojung1234!

# JWT 시크릿 키

secret-key=0b3fa07a7a9064b5153cf7390f3d4a568c481ff33bee2641b921473545b2b812

# Amazon S3

cloud.aws.credentials.access-key=AKIA6ODU6HSFGRXU2I6U

cloud.aws.credentials.secret-key=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

cloud.aws.region.static=ap-northeast-2

cloud.aws.s3.bucket=how.long.has.it.been.since.i.ate.an.orange

## 빌드하기

1) Front

flutter pub get

flutter run

flutter build apk --release --target-platform=android-arm64

2) Back-spring

Spotless Apply 실행

Clean 실행

Build실행

Bootjar 실행

## 배포하기

## 

## server {

listen 80 default\_server;

listen [::]:80 default\_server;

listen 443 ssl default\_server;

listen [::]:443 ssl default\_server;

root /var/www/html;

index index.html index.htm index.nginx-debian.html;

server\_name \_;

location / {

proxy\_pass http://localhost:8080;

proxy\_http\_version 1.1;

proxy\_set\_header Connection "";

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-Forwarded-Host $host;

proxy\_set\_header X-Forwarded-Port $server\_port;

}

ssl\_certificate /etc/letsencrypt/live/j10c203.p.ssafy.io/fullchain.pem;

ssl\_certificate\_key /etc/letsencrypt/live/j10c203.p.ssafy.io/privkey.pem;

include /etc/letsencrypt/options-ssl-nginx.conf;

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem;

}

이후 sudo systemctl restart nginx

## 서비스 이용 방법

### 카카오 로그인

준비 : 카카오로그인 API 등록

- 내 애플리케이션 → 애플리케이션 추가하기

- 테스트 앱 등록 → 필수항목 추가를 위해

- 플랫폼 등록 → 앱

### Firebase → FCM

1. Firebase 프로젝트 생성

2. 설정 → 프로젝트 설정 → 서비스 계정 → 새 비공개 키 생성

3. realtimedatabase 생성 후 주소 확인

### Amazon S3 이미지 저장

1. Amazon S3 관리 콘솔 접속
2. 버킷 만들기
3. AWS 보안 자격 증명 (IAM)
   1. 사용자 추가
   2. 권한 정책 (AmazonS3FullAccess) 추가
4. Gradle Dependency 추가
5. S3Config 클래스 작성