# 갓생살기

## 포팅 메뉴얼

🏃 실행하기

→■ 외부 서비스

### 🟋 기술 스택

- 1. 이슈관리: Jira
- 2. 형상 관리: Gitlab
- 3. 커뮤니케이션: Mattermost, Notion
- 4. 디자인: Figma
- 5. ERD: ERD cloud
- 6. 개발 환경
  - OS: Windows 10
  - DB: MariaDB 10.7.3, H2 1.4.199, Redis 3.0.504
  - Server: AWS EC2, Ubuntu 20.04, Jenkins 2.303.2, nginx 1.18.0
- 7. Frontend
  - JavaScript(ES6+), HTML 5, CSS 3
  - React 18.0.0, Redux Toolkit 1.8.1
  - TypeScript 4.6.3
- 8. Backend
  - Java JDK 11
  - Spring boot 2.6.6, Spring JPA, Spring Security
  - Lombok, Swagger 2.9.2, JWT 3.18.2
- 9. IDE & Tool
  - InteliJ 21.3.2
  - HeidiSQL
  - Visual Studio Code 1.65.2

### 📄 준비하기

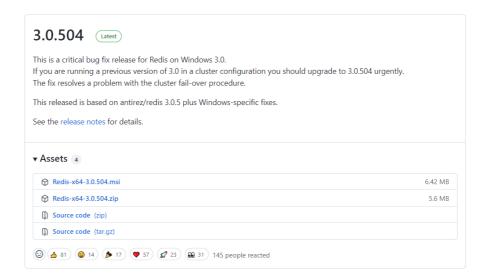
1. Git clone 받기

git clone https://lab.ssafy.com/s06-final/S06P31A405.git

- 2. 데이터 베이스 준비
- MariaDB 다운로드



#### • Redis 다운로드



#### 3. Backend application.properties 설정

```
# 로컬에서 사용할 application yml 설정 파일
# ':' 뒤에 설정값을 입력할 때는 반드시 ':' 다음에 공백이 필요
# 설정할 값들의 레벨 주의 (ex. spring.datasource.url 을 입력할 경우, datasource: 는 spring: 보다 우측으로 한 탭 이동. url: 은 datasource: 보다 우
# 기본 로그 레벨 설정
logging:
  level:
    root: info
    com.ssapy.api: debug
    org.hibernate.type.descriptor.sql: trace # trace
  profiles:
   active: local
  messages:
    basename: i18n/exception
    encoding: UTF-8
  # JWT 토큰에 사용할 secret 키값 (임의의 랜덤 키값)
  jwt:
    secret:
  # JPA 기본 설정
  jpa:
    database-platform: org.hibernate.dialect.MySQL5InnoDBDialect
    properties.hibernate:
      # 재시작 시 JPA Entity(DB 테이블 데이터)를 새로 생성할지 여부 (create:기존데이터 삭제 후 신규 생성, udpate:신규 데이터만 업데이트, none:아무 실행5
      hbm2ddl.auto: update
      format_sql: true
      show_sql: true
    use_sql_comments: true
generate-ddl: true
    open-in-view: false
    checkTemplateLocation: false
  redis:
    host: localhost
```

```
port: 6379
 # 데이터 베이스 연결 설정
 datasource:
  url: jdbc:mariadb://localhost:3306/godlife_v1?characterEncoding=utf-8&createDatabaseIfNotExist=true #데이터 베이스 이름 수정 필요
  driver-class-name: org.mariadb.jdbc.Driver
   password:
 flyway:
   enabled: false
 confia:
  activate:
    on-profile: local
   use-legacy-processing: true
 servlet:
   multipart:
    file-size-threshold: 15MB
    # 프로젝트 환경의 upload 파일을 저장할 경로
    location: C:\Develop\projects\ssafy\upload
    max-file-size: 100MB
    max-request-size: 100MB
 mvc:
   pathmatch:
    matching-strategy: ant_path_matcher
server:
 # 프로젝트 환경의 포트 설정
 port: 8000 # 포트 수정 필요
 domain: http:127.0.0.1
 servlet:
   session:
                   # 60m * 24h * 30d
    timeout: 43200m
    cookie:
      max-age: 43200m # 60m * 24h * 30d
      name: SID
      http-only: true
      secure: true
  context-path: /api
 max-http-header-size: 3145728
#aes256:
# key: WZsExuBV3GSQ55Uf
# swagger에서 테스트 할 때의 host
swagger:
 host: http://localhost:8080
# notification 부분
notification:
 mattermost:
   enabled: true # mmSender를 사용할 지 여부, false면 알림이 오지 않는다
   webhook-url: # Webhook URL을 기입
   channel: # 기본 설정한 채널이 아닌 다른 채널로 보내고 싶을 때 기입한다
   pretext: # attachments의 상단에 나오게 되는 일반 텍스트 문자
   .
color: # attachment에 왼쪽 사이드 컬러. default=red
   author-name: Back-End Exception # attachment의 상단에 나오는 이름
   footer: # attachment에 하단에 나올 부분. default=현재 시간
```

#### 4. Frontend package.json 설정

```
{
  "name": "godlife",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@emotion/react": "^11.9.0",
    "@fullcalendar/core": "^5.11.0",
    "@fullcalendar/daygrid": "^5.11.0",
    "@fullcalendar/react": "^5.11.1",
    "@mui/icons-material": "^5.6.2",
    "@mui/material": "^5.6.1",
    "@reduxjs/toolkit": "^1.8.1",
    "@testing-library/jest-dom": "^5.16.4",
    "@testing-library/react": "^13.0.1",
    "@testing-library/user-event": "^13.5.0",
    "@types/dom-to-image": "^2.6.4",
    "@types/jest": "^27.4.1",
    "@types/jest": "^27.4.1",
    "@types/react": "^18.0.4",
    "@types/react": "^18.0.4",
    "@types/react-dom": "^18.0.0",
```

```
"aws-sdk": "^2.1127.0",
"axios": "^0.26.1",
"dayjs": "^1.11.1",
"dom-to-image": "^2.6.0",
"react": "^18.0.0",
  "react-dom": "^18.0.0",
  "react-flatpickr": "^3.10.11",
  "react-gad4": "^1.4.1",
"react-google-login": "^5.2.2",
"react-hook-form": "^7.30.0",
"react-redux": "^8.0.0",
   "react-router-dom": "^6.3.0",
  "react-scripts": "5.0.1",
  "typescript": "^4.6.3",
  "util": "^0.12.4",
   "web-vitals": "^2.1.4",
  "workbox-background-sync": "^5.1.4",
"workbox-broadcast-update": "^5.1.4"
   "workbox-cacheable-response": "^5.1.4",
   "workbox-core": "^5.1.4",
   "workbox-expiration": "^5.1.4",
  "workbox-google-analytics": "^5.1.4",
"workbox-navigation-preload": "^5.1.4",
   "workbox-precaching": "^5.1.4",
  "workbox-range-requests": "^5.1.4",
"workbox-routing": "^5.1.4",
   "workbox-strategies": "^5.1.4",
  "workbox-streams": "^5.1.4"
"scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
  "eject": "react-scripts eject"
"eslintConfig": {
  "extends": [
     "react-app",
      "react-app/jest"
  1
"browserslist": {
  "production": [
     ">0.2%",
     "not dead",
      "not op_mini all"
   "development": [
     "last 1 chrome version",
     "last 1 firefox version",
      "last 1 safari version"
  1
"devDependencies": {
   "@trivago/prettier-plugin-sort-imports": "^3.2.0",
  "@types/react-flatpickr": "^3.8.6",
"@types/workbox-expiration": "^4.3.1",
   "@types/workbox-routing": "^4.3.1"
```

### 5. Frontend 모듈 다운로드

```
cd frontend/godlife

npm install
```

### 🏃 실행하기

#### 1. Backend

a. Spring boot build

```
cd backend/godlife
mvn -B -DskipTests -f backend
```

### b. 실행

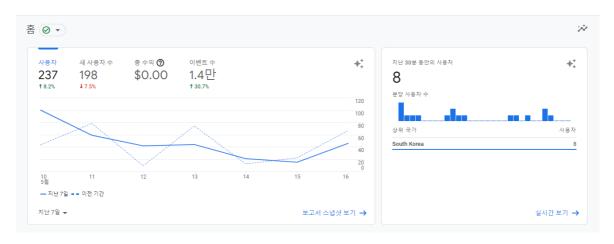
```
java -jar [filename].jar
or
mvn spring-boot:run
```

### 2. Frontend

```
cd frontend/godlife
npm start
```

### →■ 외부 서비스

### 1. Google Analytics



사용자 이용 통계를 위한 Google Analytics 사용

### 2. Social Login



Google Login과 Kakao Login 제공

### 3. **소셜 공유 기능**









- 링크 공유
- 사진 저장
- 카카오톡 공유
- 트위터 공유
- 페이스북 공유