

# 포팅메뉴얼

☰ 유형	
📅 날짜	@2024년 10월 10일

- ▼ Gitlab 소스 클론 이후 빌드 및 배포할 수 있도록 정리한 문서
- ▼ 사용한 JVM, 웹서버, WAS 제품 등의 종류와 설정 값, 버전(IDE버전 포함) 기재

## 1. JVM, 웹서버, WAS 설정 정보

- JVM
  - 버전: OpenJDK 17
  - 설정 값:

OpenJDK Runtime Environment (build 17.0.12+7-Ubuntu-1ubuntu220.04)  
OpenJDK 64-Bit Server VM (build 17.0.12+7-Ubuntu-1ubuntu220.04, mixed mode, sharing)

- 웹서버
  - 종류: Nginx
  - 버전: 1.18.0
  - 설정 파일: /etc/nginx/sites-enabled/default
  - 주요 설정 값:

```
# /etc/nginx/sites-enabled/default

server {
    listen 80 default_server;
    listen [::]:80 default_server;

    server_name j11b302.p.ssafy.io;

    # 허용할 최대 요청 크기를 설정 (예: 10MB)
    client_max_body_size 10M;

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

    location /swagger-ui {
        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;
    }

    location /v3/api-docs {
        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;
    }

    location ~ ^/(swagger|webjars|configuration|swagger-resources|v2|v3|csrf) {
        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;
    }

    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;

        set $new_cookie $http_cookie;
        if ($http_cookie ~* "(.*?;)?\s*Authorization-refresh=[^;]+(.*?)") {
            set $new_cookie $1$2;
        }
        proxy_set_header Cookie $new_cookie;
    }

    location /api/users/reissue {
        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;
    }

    location / {
        return 301 https://$host$request_uri;
    }
}

server {
    listen 443 ssl default_server;
    listen [::]:443 ssl default_server;

    server_name j11b302.p.ssafy.io;

    client_max_body_size 10M;

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

    ssl_certificate /etc/letsencrypt/live/j11b302.p.ssafy.io/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/j11b302.p.ssafy.io/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    root /var/www/html;
    index index.html index.htm index.nginx-debian.html;

    location / {
        proxy_pass $frontend_service_url;
    }

    location /ws {
        proxy_pass $frontend_service_url;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        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;
    }

    location /edit-image {
        proxy_pass http://0.0.0.0:8002;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        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_connect_timeout 300;
        proxy_send_timeout 300;
        proxy_read_timeout 300;
        send_timeout 300;

        proxy_buffer_size 128k;
        proxy_buffers 4 256k;
        proxy_busy_buffers_size 256k;
    }

    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;

        set $new_cookie $http_cookie;
        if ($http_cookie ~* "(.*?;)?\s*Authorization-refresh=[^;]+(.*)") {
            set $new_cookie $1$2;
        }
        proxy_set_header Cookie $new_cookie;
    }

    location /api/users/reissue {
        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;
    }
}

```

```

    }

    location /ssafy {
        proxy_pass https://finopenapi.ssafy.io/ssafy/;
        proxy_set_header Host finopenapi.ssafy.io;
        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 Origin http://localhost:3000;

        add_header Access-Control-Allow-Origin *;
        add_header Access-Control-Allow-Methods "GET, POST, OPTIONS";
        add_header Access-Control-Allow-Headers "Authorization, Content-Type";

        if ($request_method = OPTIONS) {
            add_header Access-Control-Allow-Origin *;
            add_header Access-Control-Allow-Methods "GET, POST, OPTIONS";
            add_header Access-Control-Allow-Headers "Authorization, Content-Type";
            add_header Access-Control-Max-Age 1728000;
            add_header Content-Type "text/plain charset=UTF-8";
            add_header Content-Length 0;
            return 204;
        }
    }
}

```

- **WAS**

- 종류: Spring Boot 내장 톨킷
- 버전: Spring Boot 3.3.3
- 설정 값:
  - 포트: 8080, 8081

▼ 빌드 시 사용되는 환경 변수 등의 내용 상세 기재

## 2. 빌드 환경 변수 및 설정

- 빌드 도구: Gradle 7.2
- build.gradle:

```

plugins {
    id 'java'
    id 'org.springframework.boot' version '3.3.3'
    id 'io.spring.dependency-management' version '1.1.6'
}

group = 'com.waterdragon.wannaeat-be'
version = '0.0.1-SNAPSHOT'

java {
    toolchain {
        languageVersion = JavaLanguageVersion.of(17)
    }
}

repositories {
    mavenCentral()
}

dependencies {
    implementation 'org.springframework.boot:spring-boot-starter-web'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
    implementation 'org.springframework.boot:spring-boot-starter-validation'
    compileOnly 'org.projectlombok:lombok'
    annotationProcessor 'org.projectlombok:lombok'
    testCompileOnly 'org.projectlombok:lombok'
    testAnnotationProcessor 'org.projectlombok:lombok'

    implementation 'net.logstash.logback:logstash-logback-encoder:7.2'

    // Security
    implementation 'org.springframework.boot:spring-boot-starter-security'
    testImplementation 'org.springframework.security:spring-security-test'
    implementation 'org.springframework.boot:spring-boot-starter-oauth2-client'

    // Database
    runtimeOnly 'com.mysql:mysql-connector-j'
    implementation 'org.springframework.boot:spring-boot-starter-data-mongodb'

    //QueryDSL
}

```

```

        implementation 'com.querydsl:querydsl-jpa:5.0.0:jakarta'
        annotationProcessor "com.querydsl:querydsl-apt:5.0.0:jakarta"
        annotationProcessor "jakarta.annotation:jakarta.annotation-api"
        annotationProcessor "jakarta.persistence:jakarta.persistence-api"

        // EmailAuth
        implementation 'org.springframework.boot:spring-boot-starter-mail'

        // CoolSMS
        implementation 'net.nurigo:sdk:4.3.0'

        // Redis
        implementation 'org.springframework.boot:spring-boot-starter-data-redis'

        // Jackson
        implementation 'com.fasterxml.jackson.datatype:jackson-datatype-jsr310'

        // QR Code - zxing
        implementation group: 'com.google.zxing', name: 'javase', version: '3.5.2'
        implementation group: 'com.google.zxing', name: 'core', version: '3.5.2'

        // Socket
        implementation 'org.springframework.boot:spring-boot-starter-websocket'
        implementation 'org.springframework.boot:spring-boot-starter-webflux'

        implementation 'org.springframework.cloud:spring-cloud-starter-aws:2.2.6.RELEASE'

        // JWT
        implementation 'io.jsonwebtoken:jjwt-api:0.11.5'
        implementation 'io.jsonwebtoken:jjwt-impl:0.11.5'
        implementation 'io.jsonwebtoken:jjwt-jackson:0.11.5'
        implementation 'com.auth0:java-jwt:4.4.0'

        // FCM
        implementation 'com.google.firebase:firebase-admin:9.1.1'

        // junit
        testRuntimeOnly 'org.junit.platform:junit-platform-launcher'

        // swagger
        implementation 'org.springdoc:springdoc-openapi-starter-webmvc-ui:2.6.0'

        // S3
        implementation 'io.awspring.cloud:spring-cloud-aws-starter-s3:3.0.0'
    }

    tasks.named('test') {
        useJUnitPlatform()
    }
}

```

- 환경 변수:

```

# 환경 변수:
# SERVER_PORT: 서버 포트 설정
# MYSQL_SERVER_ADDRESS: MySQL 서버 주소 설정
# MYSQL_DATABASE_NAME: MySQL 데이터베이스 이름 설정
# MYSQL_USERNAME: MySQL 사용자 이름
# MYSQL_PASSWORD: MySQL 사용자 비밀번호
# PHONE_AUTHCODE_EXPIRATION: SMS 인증 코드 유효기간
# QR_EXPIRATION: 비회원 식당 QR 코드 유효기간
# CART_EXPIRATION: 장바구니 유효기간
# KAKAOPAY_TID_EXPIRATION: Redis 카카오페이 TID 유효 기간
# MONGODB_USERNAME: MongoDB 사용자 이름
# MONGODB_PASSWORD: MongoDB 비밀번호
# MONGODB_HOST: MongoDB 호스트 주소
# MONGODB_PORT: MongoDB 포트
# MONGODB_DATABASE: MongoDB 데이터베이스 이름
# REDIS_SERVER_ADDRESS: Redis 서버 주소
# REDIS_SERVER_PORT: Redis 포트
# COOLSMS_API_KEY: CoolSMS API 키
# COOLSMS_SECRET_KEY: CoolSMS 비밀 키
# COOLSMS_SENDER_NUMBER: CoolSMS 발신자 번호
# SYMMETRIC_KEY: 전화번호 암호화 대칭키
# SYMMETRIC_SALT: 암호화 솔트 값
# FILE_UPLOAD_DIR: 로컬 파일 업로드 경로
# KAKAOPAY_SECRET_KEY: 카카오페이 Secret Key
# SSAFYPAY_API_KEY: 싸피페이 API 키
# FCM_SECRET_KEY_PATH: FCM Secret Key 파일 경로
# REDIRECT_URL: 리다이렉트 URL
# AWS_REGION: AWS 리전 설정
# AWS_ACCESS_KEY: AWS 액세스 키

```

```
# AWS_SECRET_KEY: AWS 시크릿 키
# AWS_S3_BUCKET: AWS S3 버킷 이름
```

▼ 배포 시 특이사항 기재

3. 배포 시 특이사항

- 배포 방식: CI/CD 파이프라인을 통해 Jenkins 및 Docker를 사용한 자동 배포
- 주의사항:
  - 배포 전 현재 실행 중인 버전의 Docker 컨테이너 종료 후 신규 버전을 실행
  - Front : 3000, 3001 Back : 8080, 8081
- 배포 후 확인사항:
  - 주요 엔드포인트를 통한 API 응답 확인
  - 서버 리소스 점검 (CPU, 메모리 사용량 등)

▼ DB 접속 정보 등 프로젝트(ERD)에 활용되는 주요 계정 및 프로퍼티가 정의된 파일 목록

```
# application.yml

server:
  port: ${SERVER_PORT} # 서버 포트 설정

logging:
  file:
    name: /var/log/spring-boot/app.log # 로그 파일 저장 경로
  level:
    root: INFO # 기본 로그 레벨

spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver # MySQL 드라이버 클래스
    url: jdbc:mysql://${MYSQL_SERVER_ADDRESS}/${MYSQL_DATABASE_NAME}?useSSL=false&serverTimezone=Asia/Seoul&characterEncoding=UTF-8
    username: ${MYSQL_USERNAME} # 데이터베이스 사용자 이름
    password: ${MYSQL_PASSWORD} # 데이터베이스 비밀번호

  jpa:
    properties:
      hibernate.dialect: org.hibernate.dialect.MySQL8Dialect # Hibernate MySQL8 방언 설정

  servlet:
    multipart:
      enabled: true
      max-file-size: 10MB # 최대 파일 크기
      max-request-size: 10MB # 최대 요청 크기

  profiles:
    include:
      - jwt
      - oauth

  phone-authcode-expiration-millis: ${PHONE_AUTHCODE_EXPIRATION} # SMS 인증 코드 유효 기간
  qr-expiration-millis: ${QR_EXPIRATION} # 비회원 식당 QR 코드 유효 기간
  cart-expiration-millis: ${CART_EXPIRATION} # 장바구니 유효 기간
  kakaopay-tid-expiration-millis: ${KAKAOPAY_TID_EXPIRATION} # 카카오페이 TID 유효 기간

  data:
    mongodb:
      uri: mongodb://${MONGODB_USERNAME}:${MONGODB_PASSWORD}@${MONGODB_HOST}:${MONGODB_PORT}/${MONGODB_DATABASE}?authSource=admin

  redis:
    host: ${REDIS_SERVER_ADDRESS} # Redis 서버 주소
    port: ${REDIS_SERVER_PORT} # Redis 서버 포트

  coolsms:
    apiKey: ${COOLSMS_API_KEY} # CoolSMS API Key
    apiSecret: ${COOLSMS_SECRET_KEY} # CoolSMS 비밀 키
    senderNumber: ${COOLSMS_SENDER_NUMBER} # CoolSMS 발신자 번호

  symmetric:
    key: ${SYMMETRIC_KEY} # 전화번호 암호화 대칭키
    salt: ${SYMMETRIC_SALT} # 암호화 솔트 값

  file:
    upload-dir: ${FILE_UPLOAD_DIR} # 로컬 파일 업로드 경로

  kakaopay:
    secret-key: ${KAKAOPAY_SECRET_KEY} # 카카오페이 Secret Key

  ssafypay:
    api-key: ${SSAFYPAY_API_KEY} # 싸피페이 API Key
```

```
firebase:
  path: ${FCM_SECRET_KEY_PATH} # FCM Secret Key 파일 경로

redirectURL: ${REDIRECT_URL} # 리다이렉트 URL

aws:
  region: ${AWS_REGION} # AWS 리전 설정
  credentials:
    accessKey: ${AWS_ACCESS_KEY} # AWS 액세스 키
    secretKey: ${AWS_SECRET_KEY} # AWS 시크릿 키
  s3:
    bucket: ${AWS_S3_BUCKET} # S3 버킷 이름
```

▼ 프로젝트에서 사용하는 외부 서비스 정보를 정리한 문서

## 1. 카카오 OAuth 로그인

- **사용 목적:** 카카오 계정을 통한 간편 로그인 기능
- **주요 엔드포인트:**
  - **토큰 발급:** <https://kauth.kakao.com/oauth/token>
  - **유저 정보 조회:** <https://kapi.kakao.com/v2/user/me>
- **설정 정보:**
  - **클라이언트 ID:** [<client\\_id>](#)
  - **리다이렉트 URI:** <https://yourdomain.com/api/auth/kakao/callback>
  - **사용 라이브러리:** [KakaoAuth SDK](#) (사용 중인 경우)
- **비고:** OAuth 인증은 클라이언트 비밀 키를 안전하게 저장하고, 토큰 만료 시 리프레시 토큰을 활용

## 2. 구글 OAuth 로그인

- **사용 목적:** 구글 계정을 통한 간편 로그인 기능
- **주요 엔드포인트:**
  - **토큰 발급:** <https://oauth2.googleapis.com/token>
  - **유저 정보 조회:** <https://www.googleapis.com/oauth2/v1/userinfo>
- **설정 정보:**
  - **클라이언트 ID:** [<client\\_id>](#)
  - **클라이언트 시크릿:** [<client\\_secret>](#)
  - **리다이렉트 URI:** <https://yourdomain.com/api/auth/google/callback>

## 3. CoolSMS (문자 전송)

- **사용 목적:** 사용자 인증 및 알림을 위한 문자 메시지 전송
- **주요 엔드포인트:**
  - **메시지 전송:** <https://api.coolsms.co.kr/sms/2/send>
- **설정 정보:**
  - **API Key:** [<api\\_key>](#)
  - **API Secret:** [<api\\_secret>](#)
  - **발신번호:** [010-xxxx-xxxx](#)
- **사용 라이브러리:** [CoolSMS SDK](#) (설치되어 있는 경우)

## 4. Firebase FCM (알림)

- **사용 목적:** FCM을 통한 푸시 알림 전송
- **주요 엔드포인트:**
  - **메시지 전송:** <https://fcm.googleapis.com/fcm/send>
- **설정 정보:**
  - **서버 키:** [<server\\_key>](#)
  - **FCM 설정 파일:** [fcm-secret-key.json](#) 파일을 서버에 배포
- **비고:** 알림 메시지 형식과 사용자 토큰 관리 주의

## 5. 카카오페이 결제 API

- **사용 목적:** 카카오페이를 통한 결제 기능
- **주요 엔드포인트:**
  - **결제 준비:** <https://kapi.kakao.com/v1/payment/ready>
  - **결제 승인:** <https://kapi.kakao.com/v1/payment/approve>
  - **결제 취소:** <https://kapi.kakao.com/v1/payment/cancel>
- **설정 정보:**
  - **Admin Key:** [<admin\\_key>](#)
  - **CID:** [<cid>](#) (상점 ID)
- **비고:** 결제 요청 시 보안 주의, 사용자의 결제 내역 관리 필요

## 6. Kakaomap API

- **사용 목적:** 지도를 활용한 위치 정보 제공
- **주요 엔드포인트:**
  - **지도 표시:** `https://dapi.kakao.com/v2/maps`
- **설정 정보:**
  - **API Key:** `<api_key>`
- **비고:** 브라우저와 서버 키 보안 주의