



최종 제출 산출물

I. 개요

1. 프로젝트 개요
2. 프로젝트 사용 도구
3. 개발환경
4. 외부 서비스
5. GitIgnore 처리한 핵심 키들

II. 빌드

1. 환경변수 형태
2. 빌드하기
3. 배포하기 (건씨 .. :^))
 - mysql
 - nginx
 - 백엔드
 - 프론트엔드
4. 외부 서비스 이용 방법
 1. Google
 2. Kakao
 3. Github
 4. FireBase

I. 개요

1. 프로젝트 개요

외출 준비를 하는 당신, 옷장을 열며 ‘입을 옷이 없네 ...’라는 생각, 자주 하지 않으셨나요? 사실 옷장엔 입을 옷이 아주 많은데 말이죠 ... 특별한 날을 기다리는 당신, 무엇을 입을지 괴로운 고민을 하고 계시지는 않는지요? 또 새 옷을 사야하나 머리가 아프지는 않으신지요?

Osiopso는 당신의 고민을 해결하기 위해 제작된 옷장 기반 코디 추천 SNS입니다. Osiopso와 함께라면, 새 옷을 사지 않고도 새로운 스타일을 도전할 수 있습니다.

2. 프로젝트 사용 도구

이슈 관리 : JIRA

형상 관리 : GitLab

커뮤니케이션 : Notion, Discord, Mattermost, KakaoTalk, Google Sheet

디자인 : Figma

UCC : 모바비

CI/CD : Gunkins

3. 개발환경

VS Code : 1.74.2

IntelliJ : idealU-2022.3.1

JVM : 17.0.5 (스프링은 11로 빌드)

Node.js : 14.21.2

SERVER : AWS EC2 Ubuntu 20

DB : MySQL 8.0.32

4. 외부 서비스

Google OAuth, Kakao OAuth, Github OAuth : application.yml에 해당 내용 있음

Google SMTP : application.yml에 해당 내용 있음

FireBase Storage : util.js에 해당 내용 있음

Remove.bg : clothes-add-picture.component.jsx, clothes-add-camera.component.jsx에 해당 내용 있음

5. GitIgnore 처리한 핵심 키들

React : .env (최상단 위치)

Spring : application.yml (src/main/resources에 위치)

II. 빌드

1. 환경변수 형태

.env

```
REACT_APP_AXIOS_URL = AXIOS 요청 주소
```

application.yml

```
spring:
  main:
    allow-bean-definition-overriding: true

# MySQL DB(배포)
datasource:
  url: jdbc:mysql://MySQL_DB_주소/SCHEMA명?useSSL=false&serverTimezone=UTC&useLegacyDatetimeCode=false&allowPublicKeyRetrieval=true
  username: 유저 이름
  password: 유저 비밀번호
  driver-class-name: com.mysql.cj.jdbc.Driver

# JPA
jpa:
  show-sql: true
  hibernate:
    ddl-auto: update
  properties:
    hibernate:
      dialect: org.hibernate.dialect.MySQLInnoDBDialect
  generate-ddl: true

# Security OAuth
security:
  oauth2:
    client:
      registration:
        google:
          clientId: 클라이언트 아이디
          clientSecret: 비밀번호
          redirectUri: 연결URL
          scope:
            - email
            - profile
        github:
          clientId: 클라이언트 아이디
          clientSecret: 비밀번호
          redirectUri: 연결URL
          scope:
            - user:email
            - read:user
      kakao:
```

```

        client-id: 클라이언트 아이디
        client-secret: 비밀번호
        redirect-uri: 연결URL
        authorization-grant-type: authorization_code
        client-authentication-method: POST
        client-name: kakao
        scope:
            - profile_nickname
            - account_email
            - profile_image
        provider:
            kakao:
                authorization-uri: https://kauth.kakao.com/oauth/authorize
                token-uri: https://kauth.kakao.com/oauth/token
                user-info-uri: https://kapi.kakao.com/v2/user/me
                user-name-attribute: id

# SMTP : 이메일 인증
mail:
    host: smtp.gmail.com
    port: 587
    #465
    username: 앱 비밀번호 발급받은 google 계정
    password: 발급받은 앱 비밀번호
    properties:
        mail:
            smtp:
                starttls:
                    enable: true
                auth: true

app:

# JWT 시크릿 키
auth:
    tokenSecret: 시크릿 키
    tokenExpirationMsec: 토큰 유효기간
# Cors
cors:
    allowedOrigins:
        Cors Origin 도메인

# Server
server:
    port: 8080
    servlet:
        context-path: /api
    error:
        include-message: always

# Logger
logging:
    level:
        root: info
    # web: debug
    # org:
    # hibernate:
    # SQL: debug

# Swagger
springdoc:
    version: 1.0
    api-docs:
        path: /
    default-consumes-media-type: application/json
    default-produces-media-type: application/json
    swagger-ui:
        operations-sorter: alpha
        tags-sorter: alpha
        path: /swagger-ui.html
        disable-swagger-default-url: true
        display-query-params-without-oauth2: true
        doc-expansion: none
    paths-to-match:
        /**

```

2. 빌드하기

1) Front

```
npm i
npm start
```

2) Back-spring

```
./gradlew.bat clean build
./gradlew.bat build
```

3. 배포하기 (건씨 .. :^))

mysql

1. sudo apt install mysql-server
2. CREATE USER '아이디'@'호스트' IDENTIFIED WITH MYSQL_NATIVE_PASSWORD BY '비밀번호';
3. FLUSH PRIVILEGES;
4. CREATE DATABASE [DB이름];
5. GRANT ALL PRIVILEGES ON [DB스키마].[권한] to '아이디'@'호스트';
6. sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
7. sudo service mysql restart

nginx

1. sudo apt-get install nginx
2. sudo systemctl stop nginx
3. sudo apt-get install letsencrypt
4. sudo letsencrypt certonly --standalone -d [도메인]
5. cd /etc/letsencrypt/live/[도메인]
6. sudo ln -s /etc/nginx/sites-available/[파일명].conf /etc/nginx/sites-enabled/[파일명].conf

백엔드

Dockerfile 작성

```
# # https://velog.io/@coastby/Docker-gradle%EB%A1%9C-%EB%B9%8C%EB%93%ED%95%98%EC%97%AC-%EB%8F%84%EC%BB%A4%EB%A1%9C-%EB%B0%B0%ED%8F%
# # https://findstar.pe.kr/2022/05/13/gradle-docker-cache/
# # https://velog.io/@mooh2jj/Jenkins-pipeline-Docker-SpringBoot%ED%94%B4%EB%A1%9C%EC%A0%9D%ED%8A%B8-%EB%B0%B0%ED%8F%AC%ED%95%98%EA%B8

FROM gradle:7.4-jdk11-alpine as builder
WORKDIR /build

# 그래들 파일이 변경되었을 때만 새롭게 의존패키지 다운로드 받게함.
COPY build.gradle settings.gradle /build/
RUN gradle build -x test --parallel --continue > /dev/null 2>&1 || true

# 빌더 이미지에서 애플리케이션 빌드
COPY . /build
RUN gradle clean build -x test --parallel

# APP
FROM openjdk:11-jdk-slim
WORKDIR /build

# 빌더 이미지에서 jar 파일만 복사
COPY --from=builder /build/build/libs/osiopso-0.0.1-SNAPSHOT.jar .

EXPOSE 8080

# root 대신 nobody 권한으로 실행
# USER nobody
ENTRYPOINT [
```

```

"java",
"-jar",
"-Djava.security.egd=file:/dev/./urandom",
"-Dsun.net.inetaddr.ttl=0",
"osiopso-0.0.1-SNAPSHOT.jar"
]

```

`docker build -t [태그명] .`

`docker run --network [네트워크] --name [컨테이너명] -d [태그명]`

프론트엔드

Dockerfile 작성

```

FROM node:14-alpine

RUN mkdir /app
WORKDIR /app

ENV PATH /app/node_modules/.bin:$PATH

COPY . /app
RUN npm install

CMD ["npm", "start"]

```

`docker build -t [태그이름] .`

`docker run --network [네트워크이름] --name[컨테이너이름] -d [태그이름]`

4. 외부 서비스 이용 방법

1. Google

OAuth 기반 소셜 로그인 API 제공

<https://developers.google.com/identity/sign-in/web/sign-in?hl=ko>

2. Kakao

OAuth 기반 소셜 로그인 API 제공

<https://developers.kakao.com/docs/latest/ko/kakaologin/rest-api>

3. Github

OAuth 기반 소셜 로그인 API 제공

<https://docs.github.com/en/apps/creating-github-apps/authenticating-with-a-github-app/identifying-and-authorizing-users-for-github-apps#2-users-are-redirected-back-to-your-site-by-github>

4. Firebase

1. Firebase 콘솔 프로젝트 만들기
2. 설정 → 프로젝트 설정 → 서비스 계정 → 설정 파일 복사
3. util.js에 내용 삽입