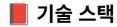# 포팅매뉴얼

📕 기술 스택

Backend

- InteliJ - 2021.2.4 Ultimate
    - Gradle, Groovy, JDK 11.0.13
    - dependencies in build.gradle

```groovy
plugins {
    id 'java'
    id 'org.springframework.boot' version '2.7.7'
    id 'io.spring.dependency-management' version '1.0.15.RELEASE'
    id "com.ewerk.gradle.plugins.querydsl" version "1.0.10"
}

group = 'com.ssafy'
version = '0.0.1-SNAPSHOT'
sourceCompatibility = '11'

configurations {
    // logback 의존성 제거 => log4j2로 변경을 위함
    all {
        exclude group: 'org.springframework.boot', module: 'spring-boot-starter-logging'
    }

    compileOnly {
        extendsFrom annotationProcessor
    }
}

repositories {
    mavenCentral()
}

dependencies {

    // 썸네일 라이브러리
    implementation 'org.jcodec:jcodec:0.2.3'
    implementation 'org.jcodec:jcodec-javase:0.2.3'

    // AWS S3 추가
    implementation 'org.springframework.cloud:spring-cloud-starter-aws:2.2.6.RELEASE'

    // log4j2 추가, 쿼리로그 추가
    implementation 'org.springframework.boot:spring-boot-starter-log4j2'
    implementation 'org.bgee.log4jdbc-log4j2:log4jdbc-log4j2-jdbc4.1:1.16'

    // MySQL
    runtimeOnly 'com.mysql:mysql-connector-j'

    // Spring Data JPA
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'

    // Spring Web
    implementation 'org.springframework.boot:spring-boot-starter-web'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'

    // Swagger 3
    implementation "io.springfox:springfox-boot-starter:3.0.0"
    implementation "io.springfox:springfox-swagger-ui:3.0.0"

    // Lombok
    compileOnly 'org.projectlombok:lombok'
    annotationProcessor 'org.projectlombok:lombok'

    // Spring Boot Devtools
    developmentOnly 'org.springframework.boot:spring-boot-devtools'
```

```
    // Spring Security
    implementation 'org.springframework.boot:spring-boot-starter-security'
    implementation 'org.springframework.security:spring-security-core:5.7.3'
    implementation 'org.springframework.security:spring-security-web:5.7.3'
    implementation 'org.springframework.security:spring-security-config:5.7.3'

    // OAuth2
    implementation 'org.springframework.boot:spring-boot-starter-oauth2-client'

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

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

    // Jwt
    implementation group: 'io.jsonwebtoken', name: 'jjwt-api', version: '0.11.2'
    runtimeOnly group: 'io.jsonwebtoken', name: 'jjwt-impl', version: '0.11.2'
    runtimeOnly group: 'io.jsonwebtoken', name: 'jjwt-jackson', version: '0.11.2'

    // QueryDsl
    implementation "com.querydsl:querydsl-jpa:5.0.0"
    implementation "com.querydsl:querydsl-apt:5.0.0"
}

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

// Qtype 생성 경로
def querydslDir = "$buildDir/generated/querydsl"
querydsl {
    jpa = true
    querydslSourcesDir = querydslDir
}
sourceSets {
    main.java.srcDir querydslDir
}
configurations {
    compileOnly {
        extendsFrom annotationProcessor
    }
    querydsl.extendsFrom compileClasspath
}
compileQuerydsl{
    options.annotationProcessorPath = configurations.querydsl
}
```

- SpringBoot - 2.7.8

- Redis - latest version in docker

Frontend

- Node.js 18.12.1

```
"@emotion/react": "^11.10.6",
    "@emotion/styled": "^11.10.6",
    "@fortawesome/free-brands-svg-icons": "^6.4.0",
    "@fortawesome/free-regular-svg-icons": "^6.4.0",
    "@fortawesome/free-solid-svg-icons": "^6.4.0",
    "@fortawesome/react-fontawesome": "^0.2.0",
    "@mediapipe/camera_utils": "^0.3.1620247984",
    "@mediapipe/drawing_utils": "^0.3.1620248257",
    "@mediapipe/face_detection": "^0.4.1646425229",
    "@mediapipe/face_mesh": "^0.4.1633559619",
    "@mediapipe/hands": "^0.3.1626903359",
    "@mediapipe/holistic": "^0.4.1627346628",
    "@mediapipe/selfie_segmentation": "^0.1.1622680987",
    "@mui/icons-material": "^5.11.16",
    "@mui/material": "^5.11.14",
    "@nivo/bar": "^0.79.1",
    "@nivo/bump": "^0.79.1",
    "@nivo/core": "^0.79.0",
    "@nivo/stream": "^0.79.1",
    "@tensorflow-models/blazeface": "^0.0.7",
    "@tensorflow/tfjs": "^4.2.0",
    "@testing-library/jest-dom": "^5.16.2",
```

```
    "@testing-library/react": "^12.1.3",
    "@testing-library/user-event": "^13.5.0",
    "aos": "^2.3.4",
    "aws-sdk": "^2.1348.0",
    "axios": "^1.3.4",
    "face-api.js": "^0.22.2",
    "framer-motion": "^6.2.8",
    "fs": "0.0.1-security",
    "html2canvas": "^1.4.1",
    "jspdf": "^2.5.1",
    "moment": "^2.29.4",
    "prop-types": "^15.8.1",
    "react": "^17.0.2",
    "react-audio-voice-recorder": "^1.0.5",
    "react-diff-viewer": "^3.1.1",
    "react-dom": "^17.0.2",
    "react-media-recorder": "^1.6.6",
    "react-quill": "^2.0.0",
    "react-router-dom": "^6.2.2",
    "react-scripts": "5.0.0",
    "react-video-recorder": "^3.19.4",
    "react-webcam": "^6.0.1",
    "recoil": "^0.7.7",
    "recoil-persist": "^4.2.0",
    "sass": "^1.60.0",
    "styled-components": "^5.3.9",
    "terminal-image": "^2.0.0",
    "video-converter": "^1.0.0",
    "video-to-audio": "^1.0.1",
    "web-vitals": "^2.1.4"
```

AI

Infra

Server

- AWS EC2 - ubuntu os 20.04 LTS
- Docker
- Nginx
  - backend , frontend 포워딩 분기 처리
- https
  - letsencrypty 인증서 발급(j8b304.p.ssafy.io)(http://j8b304.p.ssafy.io/)

CI/CD

- Jenkins
  - pipeline 스크립트 작성으로 빌드 명령 자동화
  - webhook을 이용한 빌드 유발 자동 배포
- GitLab

Cooperation

- Jira
- Notion
- MatterMost

- Webex

🌺 환경 변수

application.yml

```
spring:
  profiles:
    include: app, connection
```

application-connection.yml

```
server:
  servlet:
    context-path: /app
    port: 8080


logging:
  config: classpath:log4j2.xml

spring:

  servlet:
    multipart:
      file-size-threshold: 0MB
      location:
      max-file-size: 100MB
      max-request-size: 100MB

  security:
    oauth2:
      client:
        registration:
          google:
            client-name: google
            client-id: ${oauth2.google.id}
            client-secret: ${oauth2.google.secret}
            redirect-uri: "{baseUrl}/oauth2/callback/{registrationId}"
            authorization-grant-type: authorization_code
            scope:
              - email
              - profile
          naver:
            client-name: naver
            client-id: ${oauth2.naver.id}
            client-secret: ${oauth2.naver.secret}
            redirect-uri: "{baseUrl}/oauth2/callback/{registrationId}"
            authorization-grant-type: authorization_code
            scope:
              - email
              - name
              - profile_image
          kakao:
            client-name: kakao
            client-id: ${oauth2.kakao.id}
            client-secret: ${oauth2.kakao.secret}
            redirect-uri: "{baseUrl}/oauth2/callback/{registrationId}"
            authorization-grant-type: authorization_code
            scope:
              - account_email
              - profile_nickname
              - profile_image
            client-authentication-method: POST
        provider:
          google:
            user-name-attribute: sub
          kakao:
            authorization-uri: https://kauth.kakao.com/oauth/authorize
```

```
                user-info-uri: https://kapi.kakao.com/v2/user/me
                token-uri: https://kauth.kakao.com/oauth/token
                user-name-attribute: id
              naver:
                authorization-uri: https://nid.naver.com/oauth2.0/authorize
                user-info-uri: https://openapi.naver.com/v1/nid/me
                token-uri: https://nid.naver.com/oauth2.0/token
                user-name-attribute: response

  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://${mysql.port}/PERSONA?allowPublicKeyRetrieval=true&useSSL=false&useUniCode=yes&characterEncoding=utf8&serverTime
    username: ${mysql.user}
    password: ${mysql.pwd}

  mvc:
    pathmatch:
      matching-strategy: ant_path_matcher

  jpa:
    database: mysql
    hibernate:
      ddl-auto: update
    show-sql: true
    properties:
      hibernate:
        format_sql: true

  redis:
    host: ${redis.host}
    port: ${redis.port}
    password: ${redis.pwd}
```

application-app.yml

```
port, 시크릿키, S3, CORS allow URL등 환경변수 상세/gitignore될 정보들이 포함

app:
  oauth2:
    authorized-redirect-uris:
      -
      -


  auth:
    tokenSecret:
    tokenExpirationMsec:

mysql:
  port:
  user:
  pwd:

redis:
  host:
  port:
  pwd:

oauth2:
  google:
    id:
    secret:
  naver:
    id:
    secret:
  kakao:
    id:
    secret:

cors:
  allowedOrigins:
    -
    -

cloud:
  aws:
    stack:
      auto: false
    credentials:
```

```
    access-key:
    secret-key:
  s3:
    bucket:
  region:
    static: ap-northeast-2
```

🔍 구동 상세

젠킨스로 자동배포

🖥️ 외부 서비스

OAuth2 - Google, Naver, Kakao

OAuth2 운용 전용 계정 생성 → 소셜 사이트 앱 등록 → 유저 개인정보 활용 범위 설정 →
Redirect URL 설정 → 앱 상태 변경 및 테스팅 상태일 시 테스팅 사용자 이메일 추가 →
Spring Boot Dependency에 환경변수 및 시크릿 키 설정