포팅 매뉴얼

프로젝트 사용 도구

이슈 관리 : JIRA

형상 관리 : Gitlab, Gerrit

커뮤니케이션 : Notion, Mattermost

디자인 : Figma

UCC : CANVA, Typecast

CI/CD : Jenkins, Nginx, docker, AWS EC2, Gitlab + Jira

개발환경

IntelliJ : IntelliJ IDEA 2023.3.2

JVM : java version "17.0.9" 2023-10-17 LTS

SpringBoot : ‘3.2.2’, gradle, lombok

JWT(Json Web Token) : ???

Spring Boot Security : ???

(스웨거)Open API 3.0 : springdoc-openapi-starter-webmvc-ui : 2.2.0

Querydsl : querydsl-jpa : 5.0.0:jakarta

Android(Kotlin) : Android Studio Hedgehog, Java 1.8 target SDK 최소 24

MariaDb : mariadb-java-client, version: ‘3.3.2’

SERVER : AWS EC2 Ubuntu

Amazon S3 : aws-java-sdk-s3, version : ‘1.12.651’

외부 서비스

Naver OAuth : application.yml에 해당 내용 있음

Kakao OAuth : application.yml에 해당 내용 있음

Kakao Navigation API :

Gitignore 처리한 핵심 키들

환경 변수 형태

echo 명령어로 찾아 쓴다.

`DB_PASSWORD` = 0125

`AWS_ACCESS_KEY_ID` = AKIAU6GDYJBHU62CFVA7

`AWS_SECRET_ACCESS_KEY` = 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

`ENV` `DB_URL =` `jdbc:mariadb://i10d102.p.ssafy.io:3306/final`

`ENV` `DB_USERNAME =` `d102`

`ENV` `DB_PASSWORD =` `0125`

`ENV` `GMAIL_USERNAME =` `ukyoung147@gmail.com`

`ENV` `GMAIL_PASSWORD =` `vfkcpxxqwlyrldll`

`ENV` `JWT_SECRET_KEY =` `a3f62325a07a3194531c43a4fb97e85d8c635959ff92bf833bd86a98a5c2cbe02a3f082a5058712221c8e7cac9ad5143ed3f6d3bce64ef052b2df2bd95a0132cENV` `KAKAO_CLIENT_ID` `4748a59e12003074b3681b074013b84d`

`ENV` `NAVER_CLIENT_ID =` `ISnINdg5vxbmmrhbZ2rJ`

`ENV` `AWS_ACCESS_KEY_ID =` `AKIAU6GDYJBHU62CFVA7`

`ENV` `AWS_SECRET_ACCESS_KEY =` `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`

application.yml

```
spring:
  datasource:
    url: ${DB_URL}
    username: ${DB_USERNAME}
    password: ${DB_PASSWORD}
    driver-class-name: org.mariadb.jdbc.Driver
  jpa:
#    hibernate:
#      ddl-auto: create
    properties:
      hibernate:
        #show_sql: true
        format_sql: true
  mail:
    host: smtp.gmail.com
    port: 587
    username: ${GMAIL_USERNAME}
    # password는 앱 비밀번호 값이며 기존 구글 비밀번호와는 다른 값임
    password: ${GMAIL_PASSWORD}
    properties:
      mail:
        smtp:
          auth: true
          starttls:
            enable: true
        debug: true
        mime:
          charset: UTF-8
        transport:
          protocol: smtp

logging.level:
  org.hibernate.SQL: debug
  level:
    root: INFO
#  org.hibernate.type: trace

jwt:
  secret-key: ${JWT_SECRET_KEY}

file:
  max-file-size: 5MB

oauth:
  kakao:
    client-id: ${KAKAO_CLIENT_ID}
    url:
      auth: https://kauth.kakao.com
      api: https://kapi.kakao.com
  naver:
    client-id: ${NAVER_CLIENT_ID}
    url:
      auth: https://nid.naver.com
      api: https://openapi.naver.com

cloud:
  aws:
    credentials:
      access-key: ${AWS_ACCESS_KEY_ID}
      secret-key: ${AWS_SECRET_ACCESS_KEY}
    region:
      static: ap-northeast-2

app:
  awsServices:
    bucketName: ssafyd102

springdoc:
  packages-to-scan: com.ssafy.dmobile
  default-consumes-media-type: application/json;charset=UTF-8
  default-produces-media-type: application/json;charset=UTF-8
  swagger-ui:
    # alpha: 알파벳 순 태그 정렬, method: HTTP Method 순 정렬
    tags-sorter: alpha
    operations-sorter: method
  api-docs:
    path: /api-docs/json
    groups:
      enabled: true
  cache:
    disabled: true
```

프론트 쪽 환경변수 설정 파일

NAVER_CLIENT_ID = "ISnINdg5vxbmmrhbZ2rJ"
NAVER_CLIENT_SECRET = "Zc1XDYhfgo"
NAVER_CLIENT_NAME = "dMoblie"
KAKAO_APP_KEY = "6bc61f45686049a50d5900bd5a6c330b"

SERVER_URL = "<http://i10d102.p.ssafy.io:8080/>"

빌드하기

1. Front : Android Studio Hedgehog 실행 버튼

2. Back-spring :
    a. Jenkins credentials 설정
        - gitlab에서 personal access tokens 발급
        - Jenkins 관리 → System Configuration → System
        - Jenkins Location 입력

    b. Jenkins과 Gitlab Repository 연결
        - Credentials ADD 한 후 Username에 gitlab id, password에는 access token 추가
            - 유저네임 : git-lab id
            - password : git-lab PAT
        - script path Jenkinsfile이 존재하는지 확인

배포하기

Nginx 설정 nginx.conf

```
events {
        worker_connections 768;
}

http {
        sendfile on;
        tcp_nopush on;
        tcp_nodelay on;
        keepalive_timeout 65;
        types_hash_max_size 2048;
        include /etc/nginx/mime.types;
        default_type application/octet-stream;

        server {
                listen 80;
                server_name temp;

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

        server {
                listen 443 ssl;
                server_name https_temp;
                ssl_certificate /etc/ssl/certificate.crt;
                ssl_certificate_key /etc/ssl/private.key;
                location / {
                        proxy_pass http://i10d102.p.ssafy.io:8080;
                }
        }
        ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
        ssl_prefer_server_ciphers on;
        access_log /var/log/nginx/access.log;
        error_log /var/log/nginx/error.log;
        gzip on;
        include /etc/nginx/conf.d/*.conf;
        include /etc/nginx/sites-enabled/*;
}
```

nginx/sites-enabled

```
server
        listen 443 ssl;
        server_name i10d102.p.ssafy.io;
        ssl_certificate /etc/ssl/certificate.crt;
        ssl_certificate_key /etc/ssl/private.key;

        root /var/www/html;
        index index.html index.htm index.nginx-debian.html;
        location /api/ {
                proxy_pass http://i10d102.p.ssafy.io:8080/;
                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-Port $server_port;
                proxy_set_header X-Forwarded-Prefix /api;
        }
    }
}
```

이후 sudo service nginx start

EC2 세팅

Docker

Jenkins

S3

MariaDB

Zero SSL

배포 특이사항

프록시 설정이 /api/로 되어있어 ec2주소 뒤에 /api/를 붙여줘야 한다.

서비스 이용 방법

어플리케이션 Apk 설치