

포팅 매뉴얼

1. Gitlab 소스 클론 이후 빌드 및 배포할 수 있도록 정리한 문서

1-1. 기술 스택

FE 스택 & 버전

Kotlin	1.9.0
--------	-------

BE 스택 & 버전

JAVA	17.0.8
Spring Boot	3.1.3
JPA	3.1.3
AWS	Ubuntu 20.04 LTS
Docker	24.0.6
MariaDB	11.1.2
MongoDB	5.0.21
gradle	8.2.1
jenkins	2.414.1

1-2. 빌드 파일

FE 빌드시 사용되는 환경변수 및 설정 파일

BE 빌드시 사용되는 환경변수 및 설정 파일

- application-dev.yml

```
spring:
  main:
    allow-bean-definition-overriding: true
  datasource:
    driver-class-name: org.mariadb.jdbc.Driver
    url: [MariaDB의 url]
    username: [MariaDB의 username]
    password: [MariaDB의 password]
  data:
    mongodb:
      host: [MongoDB의 host]
      port: [MongoDB의 port]
      authentication-database: [MongoDB의 database 이름]
      database: [MongoDB의 database]
      username: [MongoDB의 username]
      password: [MongoDB의 password]
  jpa:
    hibernate:
      ddl-auto: update
    properties:
      hibernate:
        auto_quote_keyword: true
        format_sql: true
      jdbc:
        time_zone: Asia/Seoul
    show-sql: true

logging:
  level:
    reactor:
      netty:
        http:
          client=DEBUG

server:
  servlet:
    session:
      tracking-modes: cookie
      timeout: 30m
```

```

nhapi:
  ISCD: [nh api의 ISCD]
  ACCESS_TOKEN: [nh api의 ACCESS_TOKEN]
  BIRTH: [nh api 가입자의 생년월일]
  URL: https://developers.nonghyup.com/
  NORMAL_PROCESSING_MESSAGE: '정상처리 되었습니다.'

springdoc:
  api-docs:
    path: /airbank-api
    enabled: true
    version: openapi_3_0
  swagger-ui:
    path: /api.index
    with-credentials: true
    display-request-duration: true
  packages-to-scan: com.pangpang.airbank
  default-consumes-media-type: application/json
  default-produces-media-type: application/json
  cache:
    disabled: true

loan:
  constant:
    LOAN_THRESHOLD: 5

savings:
  constant:
    FAIL_THRESHOLD: 3

confiscation:
  constant:
    CONFISCATION_THRESHOLD: 7

```

1-3EC2 서버 환경설정

1-3-1. 서버의 방화벽 설정

```

sudo ufw allow 8080
sudo ufw allow 8000
sudo ufw allow 8081
sudo ufw allow 80
sudo ufw allow 9090
sudo ufw allow 27017
sudo ufw allow 3306

```

1-3-2. 서버에 Docker 설치

1-3-3. 서버에 Docker 이미지로 MariaDB 및 MongoDB 설치

1. Docker로 Mongo DB 설치 및 실행

```

docker pull mongo:5.0.21
docker run --name mongodb -v ~/data:/data/db -d -p 27017:27017 mongo:5.0.21

```

2. Docker로 MariaDB 설치 및 실행

```

docker pull mariadb
sudo docker run --name mariadb -d -p 3306:3306 -v ~/mariadata:/var/lib/mysql --restart=always -e MARIADB_ROOT_PASSWORD=[비밀번호] mariadb

```

1-3-4. 서버에 Nginx 설치

1. Nginx 설정 파일 (/etc/nginx/nginx.conf)

```

user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {

```

```

worker_connections 768;
# multi_accept on;
}

http {

    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    # server_tokens off;

    # server_names_hash_bucket_size 64;
    # server_name_in_redirect off;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    ##
    # SSL Settings
    ##

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
    ssl_prefer_server_ciphers on;

    ##
    # Logging Settings
    ##

    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;

    ##
    # Gzip Settings
    ##

    gzip on;

    # gzip_vary on;
    # gzip_proxied any;
    # gzip_comp_level 6;
    # gzip_buffers 16 8k;
    # gzip_http_version 1.1;
    # gzip_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss text/javascript;

    ##
    # Virtual Host Configs
    ##

    include /etc/nginx/conf.d/*.conf;
    include /etc/nginx/sites-enabled/*;

    # 백엔드 upstream 설정
    upstream backend {
        server 13.125.228.10:8080;
    }

    server {
        listen 80;
        server_name airbank.ssafy.life;

        location /jenkins {
            proxy_pass https://13.125.228.10:9090;
            proxy_set_header Host $http_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://$server_name$request_uri;
        }
    }

    server {
        listen 443 ssl http2;
        server_name airbank.ssafy.life;
    }
}

```

```
# ssl 인증서 적용하기
ssl_certificate /etc/letsencrypt/live/airbank.ssafy.life/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/airbank.ssafy.life/privkey.pem;

location / {
    proxy_pass http://backend;
    proxy_set_header Host $http_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 = /Auth.HTML {
    alias /home/jenkins/workspace/frontend/Auth.HTML;
}
}
```

2. 프로젝트에서 사용하는 외부 서비스 정보를 정리한 문서

카카오 소셜 로그인

- Kakao Developers 에 새 어플리케이션 등록
- debug hash key, release hash key 생성 후 등록
- Native app key와 Kakao Sdk 패키지를 이용하여 Kakao 서버와 통신

NH API

- NH에서 제공하는 금융API로 핀테크 서비스를 위한 플랫폼
- 계좌 생성 및 조회 기능
- 농협은행 계좌끼리의 송금 기능

3. DB 덤프 파일 최신본

- 별도 파일 첨부 (maria_ddl.sql)

4. 시연 시나리오

- 별도 파일 첨부 (시연 시나리오.pdf)