

# 빌드 및 배포

## 개발 환경

- Backend
  - IDE : IntelliJ 2022.02.01 (Ultimate), MobaXterm
  - SpringBoot : 3.1.3
  - Java 17
  - MariaDB
  - Redis
  - RabbitMQ
  - Docker
  - Jenkins

### ▼ Dockerfile

```
FROM openjdk:17-jdk-slim as builder

COPY gradlew .
COPY gradle gradle
COPY build.gradle .
COPY settings.gradle .
COPY src src

RUN chmod +x ./gradlew
RUN ./gradlew bootJar

FROM openjdk:17-jdk-slim
COPY --from=builder build/libs/*.jar app.jar
EXPOSE 8080

ARG SERVER_MODE
RUN echo "$SERVER_MODE"
ENV SERVER_MODE=$SERVER_MODE

ENTRYPOINT ["java", "-Dspring.profiles.active=${SERVER_MODE}", "-Duser.timezone=Asia/Seoul", "-jar", "/app.jar"]
```

### ▼ application.yml

```
spring:
  # OAuth2 social login 설정.
  security:
    oauth2:
      client:
        registration:
          google:
            client-id: 796025688077-hqc2e6abkjbg7bj08nhrbqbvu3k21mmb.apps.googleusercontent.com
            client-secret: 60CSPX-ZmSyjhmmIVJ6lSyGNSS6bI4Xnj4
            redirect-uri: '{baseUrl}/{action}/oauth2/code/{registrationId}'
            scope: email
          naver:
            client-id: yJzsI9ggxgPZoCCR9S0l
            client-secret: DNZZ4x_23p
            redirect-uri: '{baseUrl}/{action}/oauth2/code/{registrationId}'
            authorization-grant-type: authorization_code
            scope:
              - email
            client-name: Naver
          kakao:
            client-id: 3b07a5202438033c59f75ee9f58f8674
            client-secret: FvbJSCpBsGqEH47mxzcxCgJkKkBMnRis
            scope:
              - account_email
            client-name: Kakao
            authorization-grant-type: authorization_code
            redirect-uri: '{baseUrl}/{action}/oauth2/code/{registrationId}'
```

```

        client-authentication-method: client_secret_post

provider:
  naver:
    authorization-uri: https://nid.naver.com/oauth2.0/authorize
    token-uri: https://nid.naver.com/oauth2.0/token
    user-info-uri: https://openapi.naver.com/v1/nid/me
    user-name-attribute: response
  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

# rabbitmq 설정
rabbitmq:
  host: j9b308.p.ssafy.io
  port: 5672
  username: admin
  password: masoori

# key-value store인 redis 설정 (임시)
data:
  redis:
    host: j9b308.p.ssafy.io
    port: 1234

# main DB 인 MariaDB 설정 (임시)
datasource:
  driver-class-name: org.mariadb.jdbc.Driver
  url: jdbc:mariadb://j9b308.p.ssafy.io:5555/masoori?useUnicode=yes&characterEncoding=UTF-8&serverTimezone=Asia/Seoul
  username: ssafy_b308
  password: ssafy_b308_1

# JPA 설정
jpa:
  hibernate:
    ddl-auto: update
  properties:
    hibernate:
      default_batch_fetch_size: 100
    jdbc:
      batch_size: 10
      # use_sql_comments: true
      # show_sql: true
      format_sql: true

# 이메일 인증
mail:
  host: smtp.gmail.com
  port: 587
  username: masoori308
  password: ycpizzmpopiflpca
  properties:
    mail:
      smtp:
        auth: true
        timeout: 5000
        starttls:
          enable: true

# 휴대폰 인증
sms:
  api: NCSICXQZGOSRQDAU
  api-secret: RDSPTVRMQWG48BT1TFH014QWGTIHBMK5
  from: 01051548989

# log 레벨 설정
logging:
  level:
    com.fintech.masoori: DEBUG
    org:
      hibernate: debug
# OAuth2 redirect url 설정
app:
  oauth2:
    authorizedRedirectUri: http://localhost:3000/oauth/redirect

# jwt secret key 설정
jwt:
  key: YourewaitingforatrainAtrainthatwilltakeyoufarawayYouknowwhereyouhope

rabbitmq:

```

```

queue:
  challenge: challenge.req
  spending : spending.req
  analytics: analytics.req

# ssl 설정
#server:
#  ssl:
#    key-store: classpath:keystore.p12
#    key-store-password: ssafy
#    key-store-type: PKCS12

server:
  ec2-url: https://masoori.site

```

## ▼ application-dev.yml

```

spring:
  # OAuth2 social login 설정.
  security:
    oauth2:
      client:
        registration:
          google:
            client-id: 796025688077-hqc2e6abkjb7bj08nhrbqbvu3k21mmb.apps.googleusercontent.com
            client-secret: GOCSPX-ZmSyjhmmIVJG1SyGNS6bI4Xnj4
            redirect-uri: '{baseUrl}/{action}/oauth2/code/{registrationId}'
            scope: email
          naver:
            client-id: yJzsI9ggxgPZoCCR9S0l
            client-secret: DNZZ4x_23p
            redirect-uri: '{baseUrl}/{action}/oauth2/code/{registrationId}'
            authorization-grant-type: authorization_code
            scope:
              - email
            client-name: Naver
          kakao:
            client-id: 3b07a5202438033c59f75ee9f58f8674
            client-secret: FvbJSCpBsGqEH47mxzcxCgJkkBmnRis
            scope:
              - account_email
            client-name: Kakao
            authorization-grant-type: authorization_code
            redirect-uri: '{baseUrl}/{action}/oauth2/code/{registrationId}'
            client-authentication-method: client_secret_post

        provider:
          naver:
            authorization-uri: https://nid.naver.com/oauth2.0/authorize
            token-uri: https://nid.naver.com/oauth2.0/token
            user-info-uri: https://openapi.naver.com/v1/nid/me
            user-name-attribute: response
          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

# rabbitmq 설정
rabbitmq:
  host: j9b308.p.ssafy.io
  port: 5672
  username: admin
  password: masoori

# key-value store인 redis 설정 (임시)
data:
  redis:
    host: j9b308.p.ssafy.io
    port: 1234

# main DB 인 MariaDB 설정 (임시)
datasource:
  driver-class-name: org.mariadb.jdbc.Driver
  url: jdbc:mariadb://j9b308.p.ssafy.io:5555/masoori?useUnicode=yes&characterEncoding=UTF-8&serverTimezone=Asia/Seoul
  username: ssafy_b308
  password: ssafy_b308_1

# JPA 설정
jpa:
  hibernate:

```

```

    ddl-auto: update
  properties:
    hibernate:
      default_batch_fetch_size: 100
      jdbc:
        batch_size: 10
      #      use_sql_comments: true
      #      show_sql: true
      format_sql: true

# 이메일 인증
mail:
  host: smtp.gmail.com
  port: 587
  username: masoori308
  password: ycpizzmpopiflpca
  properties:
    mail:
      smtp:
        auth: true
        timeout: 5000
        starttls:
          enable: true

# 휴대폰 인증
sms:
  api: NCSICXQZGOSRQDAU
  api-secret: RDSPTVRMQWG48BT1TFH014QWGTIHBMK5
  from: 01051548989

# log 레벨 설정
logging:
  level:
    com.fintech.masoori: DEBUG
    org:
      hibernate: debug
# OAuth2 redirect url 설정
app:
  oauth2:
    authorizedRedirectUri: https://masoori.site/oauth/redirect

# jwt secret key 설정
jwt:
  key: YourewaitingforatrainAtrainthatwilltakeyoufarawayYouknowwhereyouhope

rabbitmq:
  queue:
    challenge: challenge.req
    spending: spending.req

# ssl 설정
server:
  ssl:
    key-store: classpath:keystore.p12
    key-store-password: ssafy
    key-store-type: PKCS12
    ec2-url: https://masoori.site

```

## ▼ 빌드 스크립트

- start-backend.sh

```

docker compose -f docker-compose-backend.yml pull

COMPOSE_DOCKER_CLI_BUILD=1 DOCKER_BUILDKIT=1 docker compose -f docker-compose-backend.yml up --build -d

docker compose -f docker-compose-workers.yml pull

COMPOSE_DOCKER_CLI_BUILD=1 DOCKER_BUILDKIT=1 docker compose -f docker-compose-workers.yml up --build -d

docker rmi -f $(docker images -f "dangling=true" -q) || true

```

- docker-compose-backend.yml

```

version: "3.8"

services:
  backend:
    container_name: backend

```

```

build:
  context: ../backend
  args:
    SERVER_MODE: dev
  networks:
    - deploy

networks:
  deploy:
    external: true

```

- docker-compose-workers.yml

```

version: "3.9"

services:
  challenge:
    container_name: challenge
    build:
      context: ../workers/challenge
  analytics:
    container_name: analytics
    build:
      context: ../workers/analytics
  spending:
    container_name: spending
    build:
      context: ../workers/spending

```

- Frontend

- IDE : VScode
- React : 18.2.0

```

"@fortawesome/fontawesome-free": "^6.4.2",
"@fortawesome/fontawesome-svg-core": "^6.4.2",
"@fortawesome/free-solid-svg-icons": "^6.4.2",
"@fortawesome/react-fontawesome": "^0.2.0",
"@mediapipe/tasks-vision": "^0.10.4",
"@react-spring/three": "^9.7.3",
"@react-spring/web": "^9.7.3",
"@react-three/drei": "^9.82.0",
"@react-three/fiber": "^8.14.2",
"@tanstack/react-query": "^4.33.0",
"@testing-library/jest-dom": "^5.17.0",
"@testing-library/react": "^13.4.0",
"@testing-library/user-event": "^13.5.0",
"@types/node": "^16.18.46",
"@types/react": "^18.2.21",
"@types/react-audio-player": "^0.13.0",
"@types/react-dom": "^18.2.7",
"@types/react-router-dom": "^5.3.3",
"@types/swiper": "^6.0.0",
"axios": "^1.5.0",
"craco-alias": "^3.0.1",
"event-source-polyfill": "^1.0.31",
"flickity": "^3.0.0",
"lottie-react": "^2.4.0",
"react": "^18.2.0",
"react-audio-player": "^0.17.0",
"react-copy-to-clipboard": "^5.1.0",
"react-dom": "^18.2.0",
"react-icons": "^4.10.1",
"react-native-toggle-element": "^2.0.1",
"react-pageflip": "^2.0.3",
"react-query": "^3.39.3",
"react-router-dom": "^6.16.0",
"react-scripts": "5.0.1",
"react-share": "^4.4.1",
"react-toastify": "^9.1.3",
"react-use-gesture": "^9.1.3",
"recoil": "^0.7.7",
"recoil-persist": "^5.1.0",
"styled-components": "^6.0.7",
"swiper": "^8.4.7",
"three": "^0.155.0",
"typescript": "^4.9.5",

```

```
"v6": "^0.0.0",  
"web-vitals": "^2.1.4"
```

---

## DB 접속 프로퍼티

- mariadb

```
url: jdbc:mariadb://j9b308.p.ssafy.io:5555/masoori?useUnicode=yes&characterEncoding=UTF-8&serverTimezone=Asia/Seoul  
username: ssafy_b308  
password: ssafy_b308_1
```

- Redis

```
docker run --name local-redis -p 1234:6379 --network redis-network -v /redis_temp:/data -d redis:alpine redis-server --appendonly yes  
# slaveof no one : 현재 슬레이브(복제)인 자신을 마스터로 만듭니다.  
local-redis:6379> slaveof no one
```