

Written by Team Hermes, 20 Nov 2022



LYRA

PORTING GUIDE

CONTENT

- 1 소개
 - 1.1 문서 소개
- 2 외부 서비스
 - 2.1 소셜로그인
 - 2.2 서버
 - 2.3 지도
- 3 환경 설정
 - 3.1 개발 환경
 - 3.2 Database Settings
 - 3.3 Application Properties
 - 3.4 SSL
 - 3.5 WebServer: Nginx
 - 3.6 ChatServer
- 4 데이터베이스 덤프파일
- 5 실행 방법
 - 5.1 Frontend
 - 5.2 Backend Docker Image
 - 5.3 Jenkins
 - 5.4 Nginx
- 6 시연 시나리오
 - 6.1 메인
 - 6.2 지도
 - 6.3 채팅
 - 6.4 마이페이지

포팅매뉴얼

Chapter 1 소개

1.1 문서 소개

이 문서는 Lyra를 포팅하는 방법에 대해 정리한 문서입니다. 빌드 및 배포를 위한 정보로 구성되어 있습니다.

Chapter 2 외부 서비스

2.1 소셜로그인

- 구글
- 카카오

2.2 서버

- EC2
- S3
 - ID: lyra0105@gmail.com
 - PW: Lyra@hanam^!3
 - accessKey: AKIAU33EAFESBOV5SWWW
 - secretKey: 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
 - bucket: lyra-hermes
- SSAFY Blockchain Network
- Firebase
 - ID: lyra0105@gmail.com
 - PW: Lyra@hanam^!3
 - google-services.json

```
{
  "project_info": {
    "project_number": "874986709816",
    "project_id": "lyra-89ebb",
    "storage_bucket": "lyra-89ebb.appspot.com"
  },
  "client": [
    {
      "client_info": {
        "mobilesdk_app_id": "1:874986709816:android:9596c62d137776539517cd",
        "android_client_info": {
          "package_name": "com.frontend"
        }
      },
      "oauth_client": [
        {
          "client_id": "874986709816-94dkah8g6ect53p49816b5n0gigpus3d.apps.googleusercontent.com",
          "client_type": 3
        }
      ],
      "api_key": [
        {
          "current_key": "AIzaSyCXBwS7FeM91NYxi12ySe_wtRRLB-XNjie"
        }
      ],
      "services": {
        "appinvite_service": {
          "other_platform_oauth_client": [
            {
              "client_id": "874986709816-94dkah8g6ect53p49816b5n0gigpus3d.apps.googleusercontent.com",
              "client_type": 3
            }
          ]
        }
      }
    }
  ]
}
```

```

    }
  }
},
"configuration_version": "1"
}

```

2.3 지도

- Google maps api

Chapter 3 환경 설정

3.1 개발 환경

- Frontend
 - Visual Studio Code
 - Android Studio
 - typescript 4.4.4
 - react-native 0.68.2
 - react 17.0.2
 - react-query 3.39.2
 - socket.io-client 4.5.3
 - axios 1.1.3
 - web3 1.8.0
- Backend
 - IntelliJ IDEA
 - Java 11
 - Gradle
 - SpringBoot 2.7.5

3.2 Database Settings

- DBMS: mariaDB 10.3.34
- Connection Name: lyra
- Hostname: k7c105.p.ssafy.io
- Port: 3307
- Username: hermes
- Password: Lyra@hanam^!3

```

# ec2 mariaDB 설치 및 서비스 시작
sudo apt-get update
sudo apt-get install mariadb-server

# 외부 접근용 계정 권한 설정
sudo mysql
mysql -u root -p

mysql> CREATE USER 'hermes'@'%' IDENTIFIED BY 'Lyra@hanam^!3';
mysql> flush privileges;
mysql> SELECT * FROM mysql.USER;
mysql> GRANT ALL PRIVILEGES ON *.* TO 'hermes'@'%' IDENTIFIED BY 'Lyra@hanam^!3';
mysql> GRANT ALL PRIVILEGES ON *.* TO 'hermes'@'localhost' IDENTIFIED BY 'Lyra@hanam^!3';
mysql> flush privileges;
mysql> SHOW GRANTS FOR 'hermes'@'%';
mysql> SHOW GRANTS FOR 'hermes'@'localhost';

```

```
# mariadb와 ec2 db연결하기
cd /etc/mysql/mariadb.conf.d/
vi 50-server.cnf → bind-address를 0.0.0.0으로 수정 저장

# Ubuntu 한국 시간 설정
sudo timedatectl set-timezone 'Asia/Seoul'
sudo systemctl restart mysqld
```

3.3 Application Properties

```
server:
  port: 8080
  servlet:
    encoding:
      charset: UTF-8
      enabled: true
      force: true

spring:
  application:
    name: Lyra
  datasource:
    driver-class-name: org.mariadb.jdbc.Driver
    # server
    url: jdbc:mariadb://k7c105.p.ssafy.io:3307/lyra?useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Seoul&zeroDateTimeBehav
    username: hermes
    password: Lyra@hanam^!3

  servlet:
    multipart:
      max-file-size: 100MB
      max-request-size: 100MB

  jpa:
    hibernate:
      ddl-auto: update
    show-sql: true
    generate-ddl: true

  profiles:
    include: oauth

cloud:
  aws:
    credentials:
      instance-profile: false
      accessKey: AKIAU33EAFESB0V5SWWW
      secretKey: 2dqWM35X5tcUGBHWPCqTrMnDrcBp69iTqmhpsuIi
    region:
      auto: false
      static: ap-northeast-2
    stack:
      auto: false

logging:
  level:
    com:
      amazonaws:
        util:
          EC2MetadataUtils: error

aws:
  s3:
    bucket: lyra-hermes
```

3.4 SSL

```
# letsencrypt 설치
sudo apt-get update
sudo apt-get install letsencrypt

# 포트 충돌 예방을 위해 nginx 정지
sudo systemctl stop nginx

# SSL pem 키 발급
sudo letsencrypt certonly --standalone -d k7c105.p.ssafy.io

# 발급되면 /etc/letsencrypt/live/k7c105.p.ssafy.io 아래 pem키 확인 가능
```

3.5 WebServer: Nginx

- etc/nginx/sites-available/default

```
upstream backend{
    ip_hash;
    server k7c105.p.ssafy.io:8080;
}

server {
    listen 80 default_server;
    listen [::]:80 default_server;
    server_name k7c105.p.ssafy.io;

    return 301 https://k7c105.p.ssafy.io$request_uri;
}

server {
    listen 443 ssl;
    listen [::]:443 ssl;

    ssl_certificate /etc/letsencrypt/live/k7c105.p.ssafy.io/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/k7c105.p.ssafy.io/privkey.pem;

    root /var/www/html;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;

    server_name _;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
    }

    location /api/ {
        proxy_pass http://backend/;
        proxy_redirect off;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    location /chat/ {
        proxy_pass http://k7c105.p.ssafy.io:4002;
    }

    location /jenkins/ {
        proxy_pass http://k7c105.p.ssafy.io:9090/;
    }

    location /blockchain/ {
        proxy_pass http://20.196.209.2:8545/;
    }
}
```

3.6 ChatServer

- ubuntu/ChatServer

```
npm start
```

4. 데이터베이스 덤프파일

4.1 /exec/Lyra_Dump.zip

5. 실행 방법

5.1 Frontend

- Debug용 apk 생성

```
# 터미널에서 bundle 파일 생성
react-native bundle --platform android --dev false --entry-file index.js --bundle-output android/app/src/main/assets/index.android.bundle
> android/assets/에 index.android.bundle 생성

# 안드로이드 스튜디오에서 빌드
android studio에서 [File] - [Open]을 통해 android 폴더 열기

Build -> Build APK
> /android/app/build/outputs/apk/debug 에 APK 파일 생성
```

• Release용 apk 생성

```
# bundle 파일 생성할 공간 확인
android/app/src/main 아래 assets 폴더가 있는지 확인하고 없다면 생성

# index.android.bundle 파일 생성
# 후에 소스 코드에 변경사항이 생긴다면 여기서부터 다시 작업
react-native bundle --platform android --dev false --entry-file index.js --bundle-output android/app/src/main/assets/index.android.bundle

# 안드로이드 스튜디오에서 빌드
android studio에서 [File] - [Open]을 통해 android 폴더 열기

좌측 하단 모서리 네모위에 커서를 올려 [Build Variants] 열기

:app의 Active Build Variants를 debug에서 release모드로 변경

[Build]에서 [Build APK] 선택

/android/app/build/outputs/apk/release 에 APK 파일 생성

# build 과정에서 에러 발생하는 경우
# :app:mergeReleaseResources (Duplicates resources) 에러인 경우
# - 해당 에러는 index.android.bundle을 다시 생성할 때마다 발생하므로
# 그때마다 동일하게 작업해주면 된다.
android/src/main/res에서 (drawable을 제외한) drawable-*로 시작하는 폴더 모두 삭제
raw라는 폴더도 삭제
```

5.2 Backend Docker Image

```
FROM openjdk:11-jdk
WORKDIR /var/jenkins_home/workspace/lyra/Backend/Lyra
EXPOSE 8080
CMD ["/gradlew", "clean", "build"]
VOLUME /tmp
ARG JAR_FILE=build/libs/app.jar
COPY ${JAR_FILE} app.jar
ENTRYPOINT ["java","-jar","app.jar"]
ENV TZ=Asia/Seoul
RUN apt-get install -y tzdata
```

5.3 Jenkins

```
docker image prune -a --force
mkdir -p /var/jenkins_home/images_tar

cd /var/jenkins_home/workspace/Lyra/backend/lyra
docker build -t spring .
docker save spring > /var/jenkins_home/images_tar/spring.tar

ls /var/jenkins_home/images_tar
```

5.4 Nginx

```
sudo systemctl nginx start
```

6. 시연 시나리오

6.1 메인

1. 전체 피드 스크롤
2. 동네 변경 시 바뀌는 피드 보여주기
3. 스트릿 클릭
4. 피드 좋아요

6.2 지도

1. 내 주위 공연 확인(범위별로)
2. 공연 상세에서 채팅 진입

6.3 채팅 (접속할 수 있는 인원은 모두 접속, 후원 데이터 상단에 띄워놓기)

1. 채팅 입력
2. 시연자 말고 다른 사람이 하트 발사 후 시연자도 하트 발사
3. 후원 누르기
4. 채팅 나가기
5. 내 채팅 생성

6.4 마이페이지

1. 좋아요&팔로우한 피드 및 내가 완수한 버스킹 보여주기
2. 후원받은&후원한 데이터 보여주기