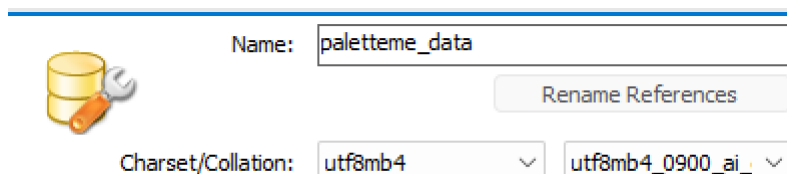


## 프로젝트 환경

- Java Version
  - 17 Oracle OpenJDK 17.0.12
- Elasticsearch Version
  - 8.13.4
- MySQL Connector(JDBC) Version
  - mysql-connector-j-8.3.0.jar
- IntelliJ

## 로컬에서 프로젝트 실행 방법

1. GitLab에서 프로젝트 다운로드
2. docker-compose.yaml 있는 위치로 이동
  - cmd창을 해당 위치에서 열기
  - docker desktop 실행해놓기
  - docker-compose up -d --build 명령어 실행
3. MySQL 설정
  - docker exec -it mysql mysql -u root -p # mysql 컨테이너 접속
  - GRANT ALL PRIVILEGES ON \*.\* TO 'ssafy'@'%' WITH GRANT OPTION; FLUSH PRIVILEGES;  
# ssafy 계정에 권한 부여
  - Schema 생성해놓기
    - 아이디: ssafy / 비밀번호: 1234
    - paletteme\_meta, paletteme\_data 라는 이름의 스키마 생성



4. Spring Boot를 실행해 테이블 만들고 데이터 넣기

- build.gradle 클릭해 프로젝트 열기
- .env 작성

```
- # Elasticsearch 비밀번호
ELASTIC_USERNAME=elastic
ELASTIC_PASSWORD=admin1234
ELASTIC_KIBANA_USERNAME=kibana_system
ELASTIC_KIBANA_PASSWORD=admin1234

# 데이터베이스 설정
MYSQL_ROOT_PASSWORD=1234
MYSQL_DATABASE=test_db
MYSQL_USER=ssafy
MYSQL_PASSWORD=1234

# 데이터소스 설정
SPRING_DATASOURCE_META_URL=jdbc:mysql://localhost:3306/paletteme_
meta
SPRING_DATASOURCE_DATA_URL=jdbc:mysql://localhost:3306/paletteme_
data

# 영단어랑 학준님은 USERNAME 에 root 쓰면 됨
SPRING_DATASOURCE_USERNAME=root
SPRING_DATASOURCE_PASSWORD=1234

# REDIS
REDIS_HOST=localhost
REDIS_PORT=6379

#s3 설정
CLOUD_AWS_S3_CREDENTIALS_ACCESS_KEY=AKIAU6GDWK43KC6YLCJW
CLOUD_AWS_S3_CREDENTIALS_SECRET_KEY=AJbcz2VYPjgcSqxJObbUjqCsA1uoJ
PhJa2DvuOhc
CLOUD_AWS_S3_REGION=ap-northeast-2
CLOUD_AWS_S3_MYPAGE_BUCKET=byeol-mypage
DEFAULT_PROFILE_IMAGE_URL=https://cdn.pixabay.com/photo/2015/10/05
/22/37/blank-profile-picture-973460_1280.png

#jwt 키
JWT_SECRET=palletemeSuperSecureJwtKey_2025!@

# SMS
SMS_API_KEY=NCSVPNQBKXUIKC63
SMS_API_SECRET=SN8ZRLICW554YTNRHZAVXAL7JVQAEBTK
SMS_SENDER=01033076019

# chatgpt
CHATGPT_API_KEY=sk-proj-
LwdqyuAoOtd3N8AJB94DnIUeoP9IhXYNZHYpj6pI_T6m_jNIK_JwDA1Av4icoqv4o
TMjFzL3ueT3BlbkFJBpTBepfJ3HGdAbkhfs_Mi85daaFF2yV3dlajVREht4_Bph26
VRCmY61_pcJevOhMvhGg0BQXkA

# phoneNumber
# 24, 16 byte. (문자수가 아닌 byte 수 기준)
AES_SECRET_KEY=palletemeZChoCoKeyX25014
AES_INIT_VECTOR=lastVitaCrAES202
```

- 프로젝트 실행 -> 자동으로 DB에 테이블 생성
- 실행 시 오류가 난다면?



- <https://saakmiso.tistory.com/180> 해결해보기
- 데이터 넣는 방법은 노션 참고!
  - <https://www.notion.so/kimhyunhee/mysql-1c963f189b7e80bdbf3bccaa6c0da17c>
  - export 폴더에 있는 artworks.csv, eras.csv, museum.csv, artists.csv 를 넣으면 된다.
- Docker Desktop에서 logstash까지 실행해주면 elasticsearch에 넣는 것까지 완료!

## 배포 시 프로젝트 설정

- localhost로 설정한 부분들은 모두 컨테이너 이름으로 변경
  - Ex) `jdbc:mysql://localhost:3306/paletteme_data`  
     ➔ `jdbc:mysql://mysql:3306/paletteme_data`
- Dockerfile, nginx 설정 파일은 모두 GitLab에서 다운받아 오기
- Spring 프로젝트 빌드
  - gradle.bat wrapper
  - gradle.bat build
- EC2에 올리기
  - `scp -i J12A505T.pem -r paletteme ubuntu@43.203.218.239:/home/ubuntu/`
- EC2 접속
- 프론트 빌드
  - npm install
  - npm run build
- docker에 필요한 요소들 띄우기
  - docker compose up -d
- Elasticsearch의 kibana\_system 계정의 비밀번호 생성

```
curl -u elastic:admin1234 -X PUT http://localhost:9200/_security/user/kibana_system/_password ₩  
-H "Content-Type: application/json" ₩  
-d '{"password": "admin1234"}'
```

9. docker restart kibana

10. 브라우저에서 kibana로 접속하기 위한 계정 생성

```
curl -u elastic:admin1234 -X POST http://localhost:9200/_security/user/yd7052 ₩  
-H "Content-Type: application/json" ₩  
-d '{  
  "password": "palettemeisbest",  
  "roles": [ "superuser" ],  
  "full_name": "yd"  
}'
```

11. Kibana에 접속해서 인덱스 템플릿 생성

A. <https://www.notion.so/kimhyunhee/1c963f189b7e8078aab7eb23a036aed7> 에서 확인

12. MySQL에 데이터 올리기

A. <https://www.notion.so/kimhyunhee/mysql-1c963f189b7e80bdbf3bcca6c0da17c> 이  
참고하여 동일하게 올리면 된다.

13. Elasticsearch에 데이터 올리기

A. docker compose up -d logstash

14. url로 접속!