**Tide 포팅 매뉴얼**

1. 개발 환경
   1. Java – Azul 11.0.18
   2. Spring Boot – 2.7.9
   3. NodeJs – 18.12.1
   4. MariaDB - 10.11.2
   5. Docker – 23.0.1
   6. Jenkins – 2.387.1
   7. Python – 3.9.16
2. 데이터 삽입
   1. sql폴더의 musicproejctdump.sql 파일을 mariadb의 music DB를 만들어 삽입
3. 네트워크 생성
   1. docker network create --gateway 172.18.0.1 --subnet 172.18.0.0/16 music-project
4. 빌드
   1. backend 폴더의 recommendService 폴더를 제외한 나머지 Spring 프로젝트를 빌드 ./gradlew clean build -x test (오류 발생 시 chmod +x gradlew 를 실행해 권한 부여
5. 도커 이미지 빌드
   1. 프론트
      1. docker build -t front-tide .
   2. DB
      1. docker pull mariadb
      2. docker run -d -p 3306:3306 --network music-network --name mymariadb -v testdb-volume:/var/lib/mysql -e MYSQL\_ROOT\_PASSWORD={DBpassword} mariadb
   3. 백엔드
      1. backend/discoveryservice
         1. docker build -t discovery-service .
         2. docker run -d -p 8761:8761 --network music-network --name discovery-service discovery-service
      2. backend/apigateway-service
         1. docker build -t apigateway-service .
         2. docker run -d -p 8000:8000 --network music-network -e "eureka.client.serviceUrl.defaultZone=http://discovery-service:8761/eureka/" --name apigateway-service apigateway-service
      3. backend/user
         1. docker build -t user .
         2. docker run -d --network music-network --name api-user -e "spring.datasource.url=jdbc:mysql://{DB주소}/music" -e "eureka.client.serviceUrl.defaultZone=http://discovery-service:8761/eureka/" api-user
      4. backend/diary
         1. docker build -t diary .
         2. docker run -d --network music-network --name api-diary -e "spring.datasource.url=jdbc:mysql://{DB주소}/music" -e "eureka.client.serviceUrl.defaultZone=http://discovery-service:8761/eureka/" api-diary
      5. backend/music
         1. docker build -t music .
         2. docker run -d --network music-network --name api-music -e "spring.datasource.url=jdbc:mysql://{DB주소}/music" -e "eureka.client.serviceUrl.defaultZone=http://discovery-service:8761/eureka/" api-music
      6. backend/recommendService
         1. docker build -t flaskapp .
         2. docker run -d -p 4000:4000 --name flaskapp flaskapp
   4. Nginx Proxy Manager
      1. nginxproxymanager 폴더로 이동 후 docker-compose up –d 실행 오류 발생 시 위치에 data, letsencrypt 폴더 생성
      2. DuckDNS를 이용해 서브 도메인 tideapi.duckdus.org, tideflask.duckdus.org, tidemusic.duckdus.org 생성
      3. {ip주소}:5000포트로 접속해 기본 아이디 [admin@example.com](mailto:admin@example.com) 비밀번호 changeme 입력해 접속 후 ssl, tideapi는 8000포트로 tidemusic은 3000포트로 tideflask는 4000포트로 리버스 프록시 설정
6. 실행
   1. <https://tidemusic.duckdns.org> 실행시 서비스 실행가능