

**D108 : Sduty**

삼성SW청년아카데미 구미캠퍼스 7기

공통 프로젝트

2022/7/11 ~ 2022/8/19

**포팅 매뉴얼**

담당 컨설턴트 : 서성수

권용준(팀장), 김남희, 김정윤, 정봉진, 최영희, 편예린

**목차**

1. 프로젝트 기술 스택…………………………………………………………….3
2. 빌드 상세내용……………………………………………………………………..4
3. 배포 특이사항……………………………………………………………………..8
4. DB 계정……………………………………………………………………………..10
5. 프로퍼티 정의……………………………………………………………………11
6. 외부 서비스……………………………………………………………………….12
7. **프로젝트 기술 스택**
   * 형상관리 : Gitlab
   * 이슈관리 : Jira
   * 프로젝트관리 : Notion
   * 커뮤니케이션 : Mattermost
   * 디자인 : Figma
   * OS : Windows 10, 11
   * Database : Mysql 8.0.28
   * language : Kotlin, Java 8
   * Front-End
     + Android
     + Kotlin 1.7.10
     + JDK 11.0.11
     + Gradle 7.2
     + SDK
       1. min 21
       2. target 31
       3. compile 32
   * Back-End
     + Spring Boot 2.7.1, Maven 3.6.3
     + Lombok 1.18.24
     + JPA, Swagger 2.X, Spring Security 2.7.1 , https
   * Server
     + AWS EC2
     + Ubuntu 20.04 LTS
     + Jenkins
   * IDE
     + MySQL Workbench 8.0 CE
     + Android Studio 2021.2.1 Patch2
     + STS 3.9.14 on Eclipse 4.17
8. **빌드 상세 내용**

**백엔드 빌드 내용**

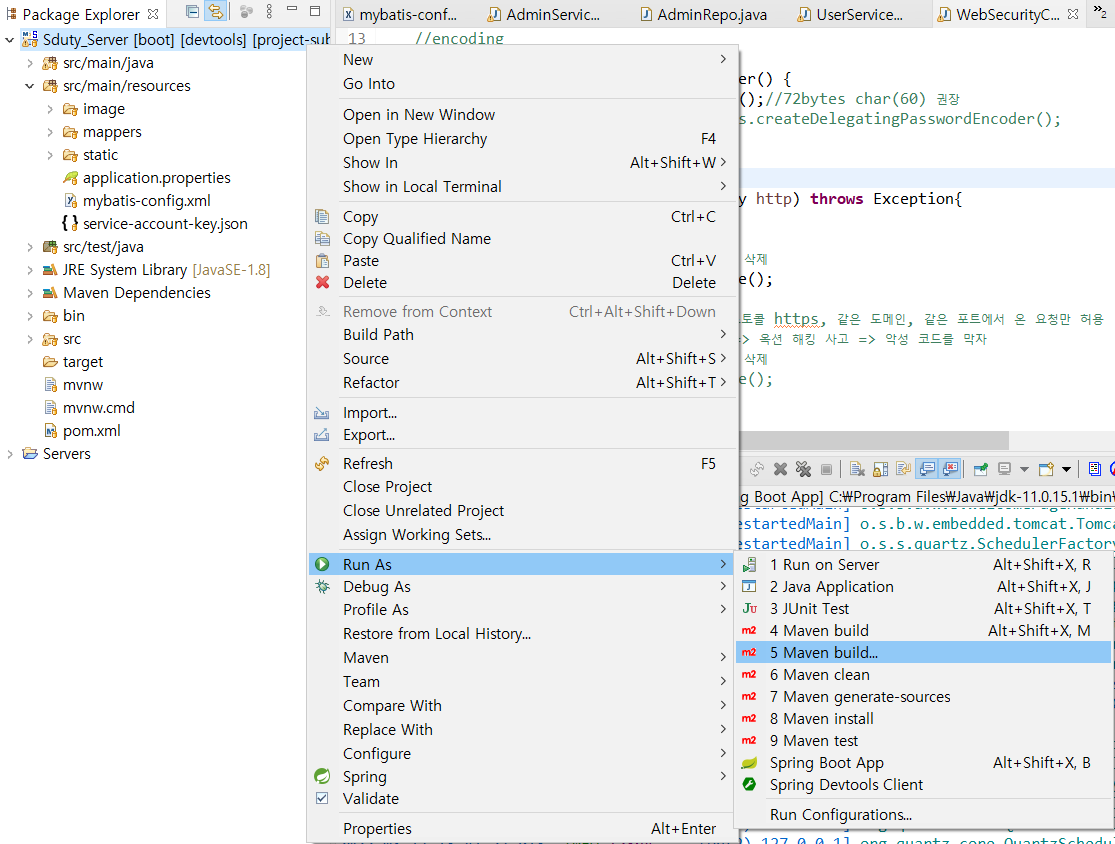
빌드를 시작하기 전에, application.properties파일에서 ssl관련 설정을 주석처리 해준다.

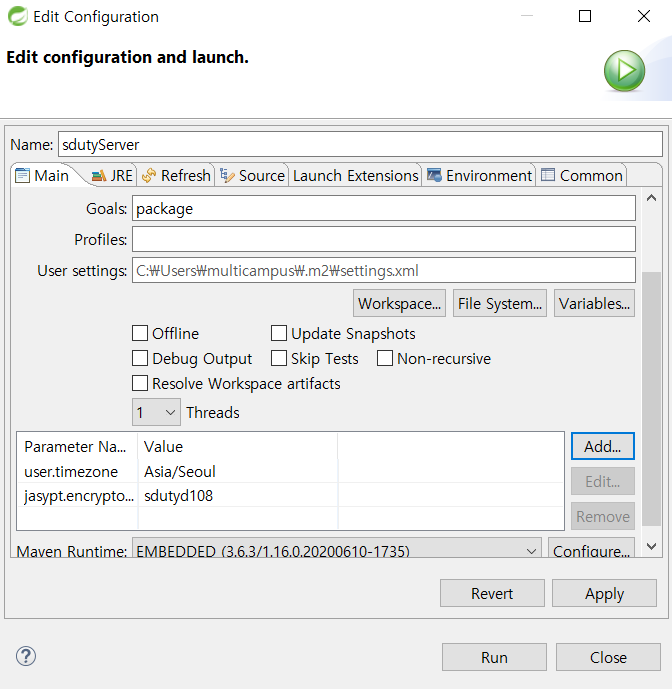
ssl을 사용하려면, domain이 필요하기 때문에 local에서 사용할 수 없기 때문이다.

****

[GUI]

프로젝트 우 클릭 => Run As => Maven build



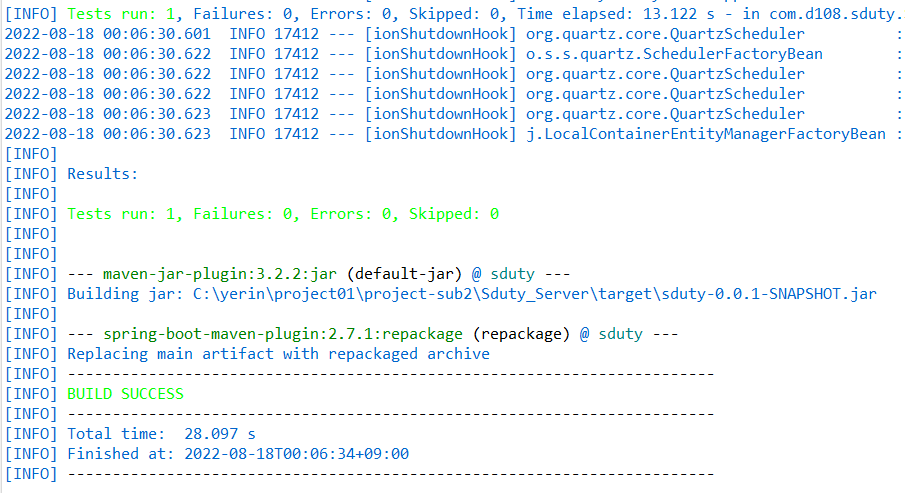


Name과 Goals를 지정한 후,

프로그램 실행 시, 외부에서 주입시켜주는 데이터를 추가시켜준다.

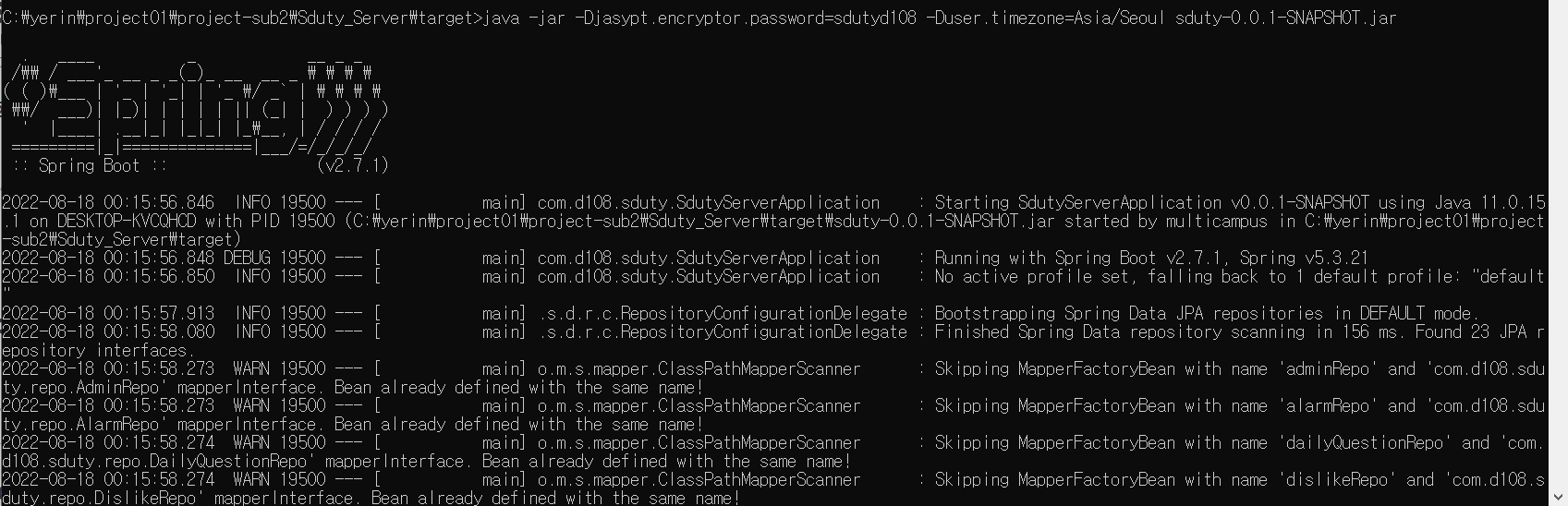
1. timezone설정
2. 암호화 키 값

설정이 끝나면 [Run] 클릭



BUILD SUCCESS가 되면, 빌드가 성공적으로 완료된 것이며, 위의 경로를 따라 target폴더에서 .jar 파일을 확인해볼 수 있다.

실행 예시



HTTPS 설정

\*미리 도메인 연결 해놓거나, SSAFY EC2 도메인 사용. 80포트는 열려 있어야 함

-도커 컴포즈 설치

docker-compose sudo curl -L "<https://github.com/docker/compose/releases/download/v2.7.0/docker-compose-windows-x86_64.exe>" -o /usr/local/bin/docker-compose

-Certbot, NginX 설치

sudo snap install core; sudo snap refresh core

sudo snap install --classic certbot

sudo ln -s /snap/bin/certbot /usr/bin/certbot

sudo apt install nginx

-키 발급

sudo certbot –nginx -> 이메일 입력, 발급받을 도메인 입력

-발급된 키 jks로 변환

cd /etc/letsencrypt/live/gumid108.kro.kr/

openssl pkcs12 -export -in fullchain.pem -inkey privkey.pem -out cert\_and\_key.p12 -name ttp -CAfile chain.pem -caname root

apt install openjdk-11-jre

keytool -importkeystore -deststorepass test12 -destkeypass test12 -destkeystore new\_key.jks -srckeystore cert\_and\_key.p12 -srcstoretype PKCS12 -srcstorepass test12

keytool -export -alias ttp -keystore new\_key.jks -rfc -file ncert.cer

keytool -import -alias t\_key -file ncert.cer -keystore t\_key.jks

new\_key.jks, t\_key.jks 를 스프링 서버 classpath로 이동.

**프론트엔드 빌드 내용**

안드로이드 스튜디오 버전 및 Setting – Plugin – Kotlin 버전 확인

Gradle-wrapper Gradle 버전 확인

1. **배포 특이사항**

배포 명령어 정리

-젠킨스 설치

sudo wget -q -O - <https://pkg.jenkins.io/debian-stable/jenkins.io.key> | sudo apt-key add -

sudo sh -c 'echo deb <https://pkg.jenkins.io/debian-stable> binary/ > \ /etc/apt/sources.list.d/jenkins.list'

sudo apt-get update

sudo apt install jenkins

sudo systemctl stop jenkins

sudo nano /etc/default/jenkins sudo nano /usr/lib/systemd/system/jenkins.service -> 접속 포트 변경

sudo systemctl start Jenkins -> 젠킨스 시작

sudo apt install maven -> Maven 설치

설정한 포트로 젠킨스 접속, 초기 비밀번호 : sudo cat /var/lib/jenkins/secrets/initialAdminPassword

-플러그인 설치 (Gitlab, SSH)

- Jenkins 관리 – Global Tool Configuration

- JDK JAVA\_HOME : /usr/lib/jvm/java-8-openjdk-amd64

- Git PATH : /usr/bin/git

- Maven : Install Automatically Maven Version 3.6.3 (버전 확인: mvn --version)

-새로운 아이템 추가

-소스 코드 관리 Git : RepositoryURL , Credential에 Gitlab ID/PW 입력

-빌드 유발 : Gitlab webhook – 고급 – Secret Token에 깃랩 리포지토리에서 발급한 token 추가

Build – Add build step 에서 Invoke top-level Maven targets 추가

* Maven Version : 젠킨스 관리에서 추가한 Maven 등록

-Goals : clean package -D maven.test.skip=true

-POM : ./Sduty\_Server/

-Excute shell 추가

cat /dev/null>/home/ubuntu/ServerLog.log #서버 로그

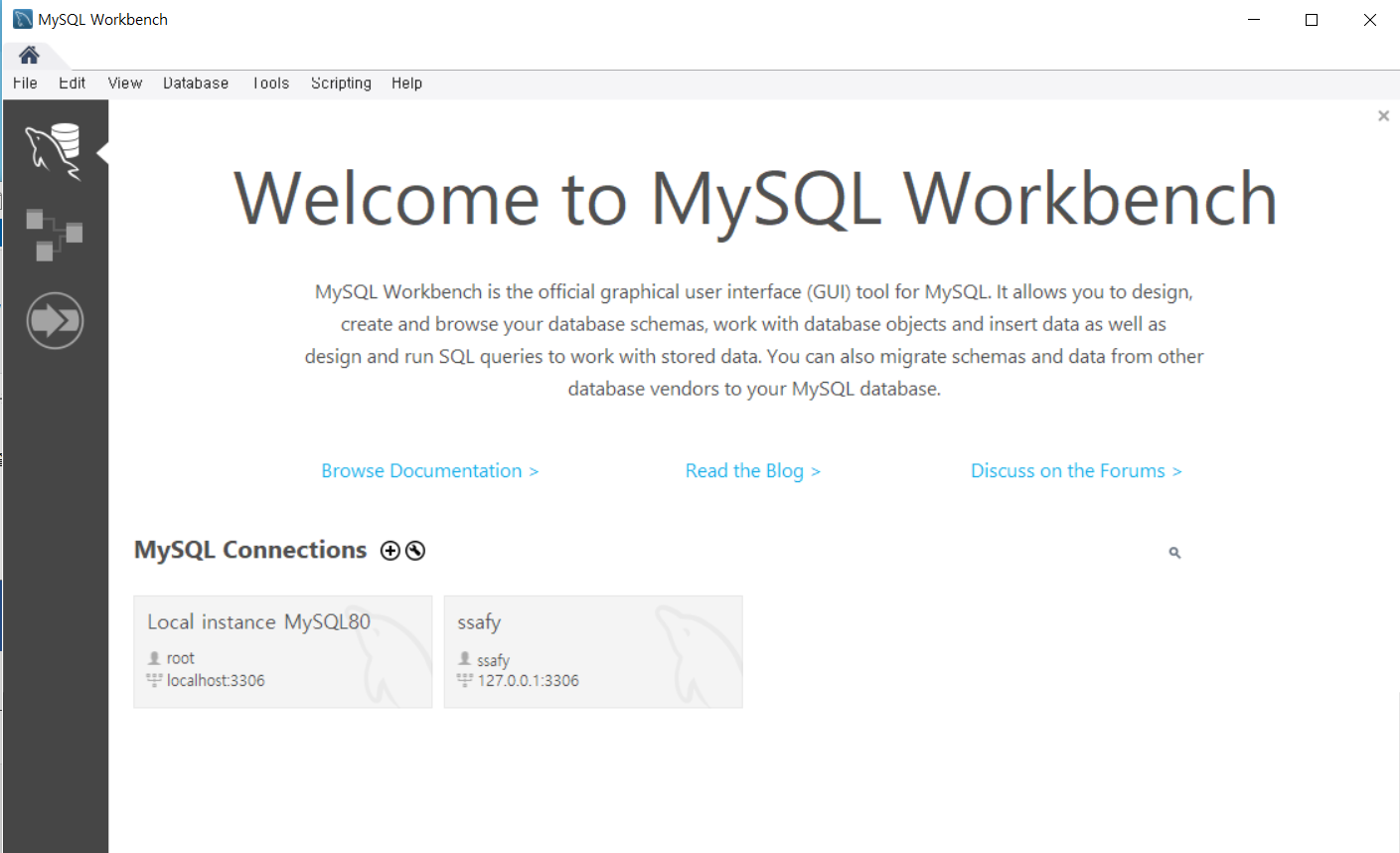
fuser -n tcp -k 8090 #서버 포트 사용중인 프로그램 끔

BUILD\_ID=dontKillMe nohup java -jar -Djasypt.encryptor.password=sdutyd108 -Duser.timezone=Asia/Seoul Sduty\_Server/target/sduty-0.0.1-SNAPSHOT.jar >> /home/ubuntu/ServerLog.log & # 서버 백그라운드로 실행

\*빌드 실패 시 권한 확인 혹은 SSH 연결 명령어 실행

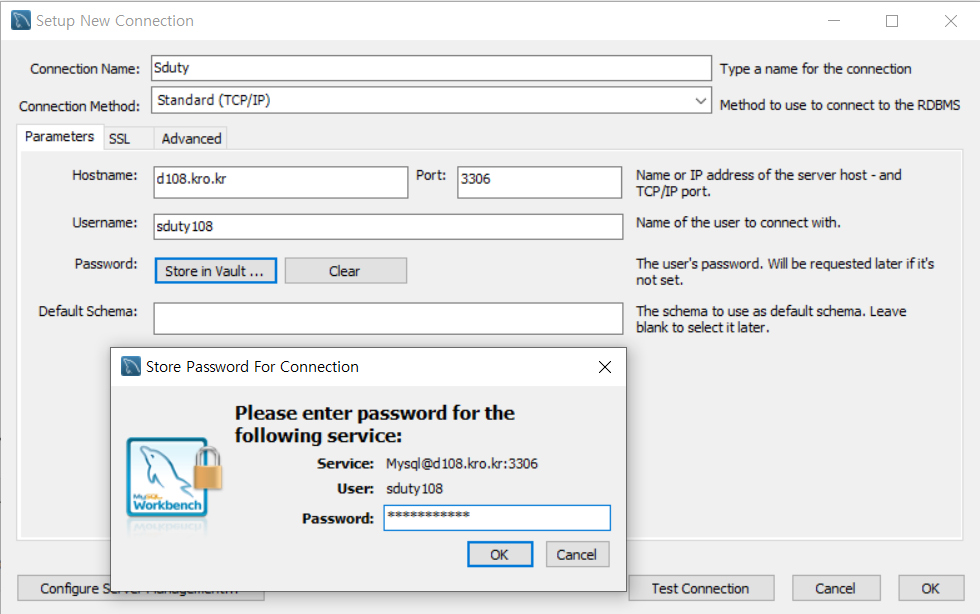
\*젠킨스로 빌드시 프로젝트 경로는 /var/lib/Jenkins/workspace/ITEM\_NAME/

1. **DB 계정**
2. MySQL Connection 추가



* MySQL Workbench를 실행합니다
* MySQL Connection을 추가하기 위해 (+) 버튼을 클릭합니다.

1. Connection 설정



* Connection Name : Sduty
* Hostname : d108.kro.kr / 3306
* Username : sduty108
* Password : sduty132765
* 위 내용 그대로 입력하고 Test Connection 을 실행하여 연결이 되는지 확인한다.

1. **프로퍼티 정의**
2. EC2 세팅

* 서버 내 필요한 프로그램 세팅을 위한 업데이트
  + sudo apt-get update
  + sudo apt-get upgrade
* MySQL설치
  + sudo apt-get install mysql
* MySQL 시간대를 한국 시간대에 맞게 설정
  + sudo vim /etc/mysql/mysql.conf.d/mysqld
  + 명령어 입력 후 가장 아랫줄에 default-time-zone="+09:00"를 추가.
  + :wq를 입력하여 vim을 저장하고 종료.
  + sudo service mysql restart로 서비스를 재시작하여 수정사항 적용.
* Java 설치
  + sudo apt-get install openjdk-8-jdk

1. **외부 서비스**

Kakao Login 설정

<https://developers.kakao.com/console/app>

-애플리케이션 추가 후 네이티브 앱 키를 AndroidManifest.xml에 추가

- 키 해시 발급 후 Android플랫폼에 package 명과 함께 추가

keytool -exportcert -alias androiddebugkey -keystore %USERPROFILE%\.android\debug.keystore -storepass android -keypass android | openssl sha1 -binary | openssl base64

\*Release용 (구글 플레이스토어 등록 시 키 해시 추가해야 함)

keytool -exportcert -alias <RELEASE\_KEY\_ALIAS> -keystore <RELEASE\_KEY\_PATH> | openssl sha1 -binary | openssl base64

-팀 관리에서 카카오 로그인 이용할 이메일 추가

Naver Login 설정

<https://developers.naver.com/apps/#/list>

-애플리케이션 등록

- 안드로이드 ApplicationClass NaverIdLoginSDK.initialize 에 ClientID, Client Secret 추가

- 멤버관리에서 네이버 로그인 이용할 아이디 추가

Solapi (SMS 발송)

<https://console.solapi.com/dashboard>

-API 키 발급 후 안드로이드 Constants.kt에 API\_KEY, API\_SECRET\_KEY 업데이트

Sendbird ( 화상회의)

<https://dashboard.sendbird.com/>

-Application Chat + Calls 로 생성 후 안드로이드 Constants.kt에 APP\_ID 업데이트