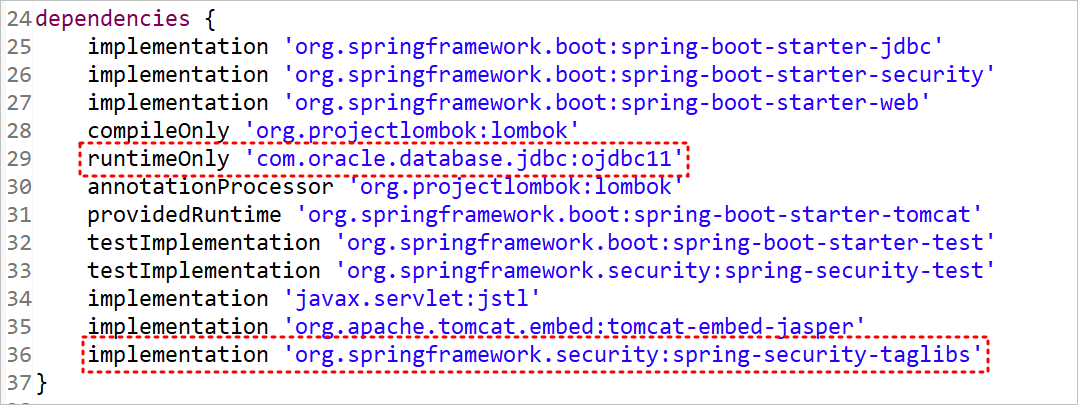
시큐리티 - JDBC 연동 및 taglib 사용

# 프로젝트명 : B09cSecurityJDBC

## 준비사항

* 기존 B09bSecurityCustomLogin 프로젝트를 복사한 후 이름을 B09cSecurityJDBC 로 변경한다.
* 컨텍스트 루트 경로와 settings.gradle 를 변경한다.
* Refresh Gradle Project 를 눌러 적용한다.

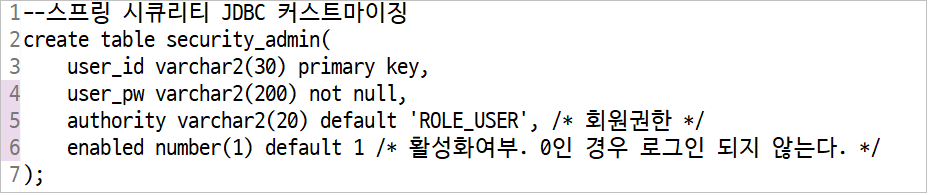
## 의존설정 추가



| implementation 'org.springframework.boot:spring-boot-starter-jdbc'  runtimeOnly 'com.oracle.database.jdbc:ojdbc11'  implementation 'org.springframework.security:spring-security-taglibs' |
| --- |

gradle을 reflash 한다.

## 테이블생성



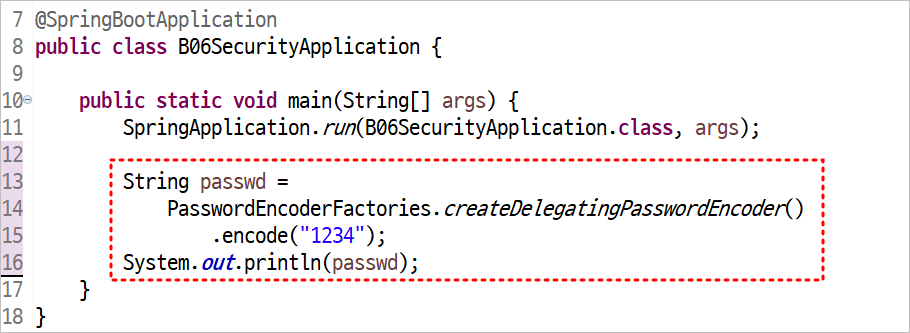
## 더미데이터 입력

| insert into security\_admin values ('user1', '1234', 'ROLE\_USER', 1);  insert into security\_admin values ('user2', '1234', 'ROLE\_USER', 0);  insert into security\_admin values ('admin1', '1234', 'ROLE\_ADMIN', 1);  insert into security\_admin values ('admin2', '1234', 'ROLE\_ADMIN', 0);  commit; |
| --- |

단, 스프링 시큐리티에서는 패스워드를 반드시 암호화 한 후 입력해야 한다.

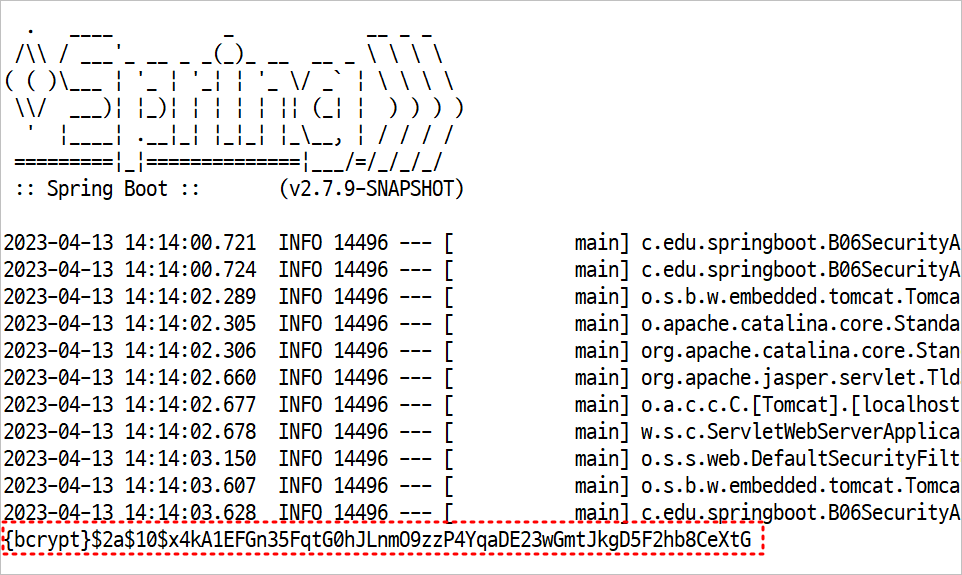
따라서 아래와 같이 처리하면 된다.

### src/main/java/com/edu/springboot/B06SecurityApplication



해당 코드를 실행할때는 우클릭 -> Run as -> Java application을 클릭하면 된다.

그러면 Java 프로그램처럼 실행되고 콘솔에서 결과를 볼 수 있다.



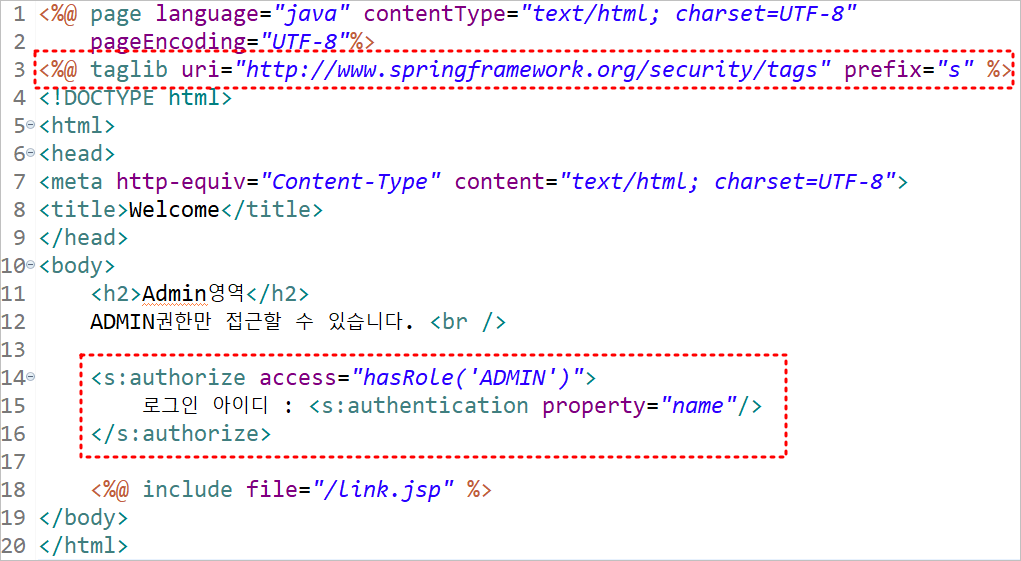
암호화된 패스워드를 복사한 후 쿼리를 수정하고 더미데이터를 입력한다.

복사할때 {bcrypt} 부분을 제외하고 복사한다.

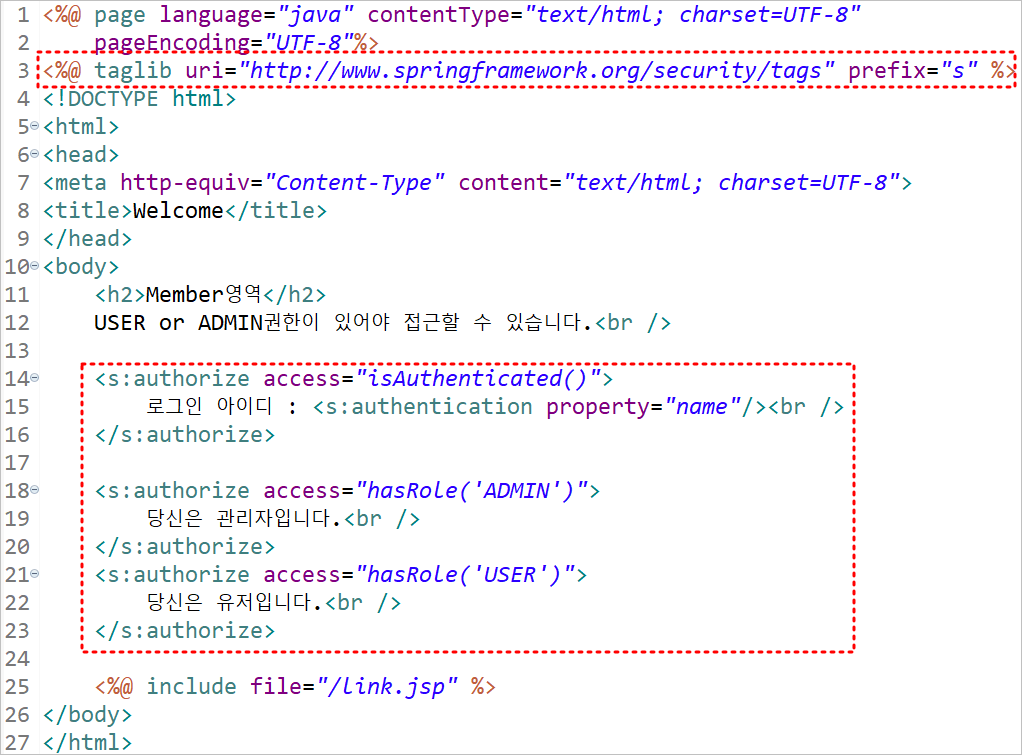
이미 입력한 상태라면 update문을 사용하면 된다.

**여기까지 작성하세요.**

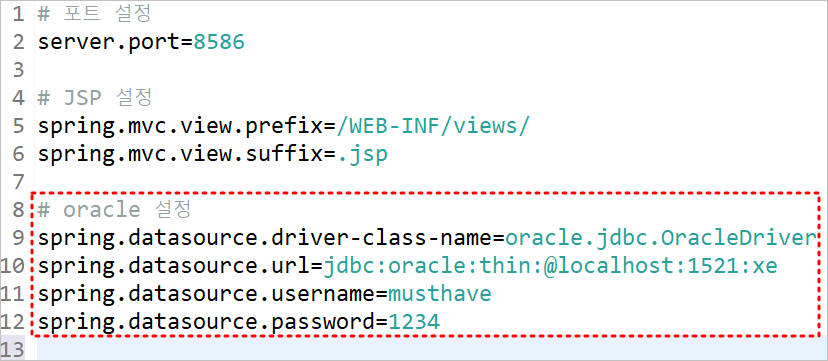
### 뷰 : webapp/WEB-INF/views/admin.jsp



### 뷰 : webapp/WEB-INF/views/member.jsp



### 프로젝트 설정 : resources/application.properties



### 시큐리티 설정 : com/edu/springboot/auth/WebSecurityConfig.java

기존 인메모리 방식을 사용했던 users(), passwordEncoder() 메서드는 주석으로 처리한다.

