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

```
dependencies {
    implementation 'org.springframework.boot:spring-boot-starter-jdbc'
    implementation 'org.springframework.boot:spring-boot-starter-security'
    implementation 'org.springframework.boot:spring-boot-starter-web'
    compileOnly 'org.projectlombok:lombok'
    runtimeOnly 'com.oracle.database.jdbc:ojdbc11'
    annotationProcessor 'org.projectlombok:lombok'
    providedRuntime 'org.springframework.boot:spring-boot-starter-tomcat'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'
    testImplementation 'org.springframework.security:spring-security-test'
    implementation 'javax.servlet:jstl'
    implementation 'org.apache.tomcat.embed:tomcat-embed-jasper'
    implementation 'org.springframework.security:spring-security-taglibs'
}
```

```
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

```
--스프링 시큐리티 JDBC 커스트마이징
create table security_admin(
    user_id varchar2(30) primary key,
    user_pw varchar2(200) not null,
    authority varchar2(20) default 'ROLE_USER', /* 회원권한 */
    enabled number(1) default 1 /* 활성화여부. 0인 경우 로그인 되지 않는다. */
);
```

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

```java
@SpringBootApplication
public class B06SecurityApplication {

    public static void main(String[] args) {
        SpringApplication.run(B06SecurityApplication.class, args);

        String passwd =
            PasswordEncoderFactories.createDelegatingPasswordEncoder()
                .encode("1234");
        System.out.println(passwd);
    }
}
```

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

그러면 Java 프로그램처럼 실행되고 콘솔에서 결과를 볼 수 있다.

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::       (v2.7.9-SNAPSHOT)

2023-04-13 14:14:00.721  INFO 14496 --- [           main] c.edu.springboot.B06SecurityA
2023-04-13 14:14:00.724  INFO 14496 --- [           main] c.edu.springboot.B06SecurityA
2023-04-13 14:14:02.289  INFO 14496 --- [           main] o.s.b.w.embedded.tomcat.Tomca
2023-04-13 14:14:02.305  INFO 14496 --- [           main] o.apache.catalina.core.Standa
2023-04-13 14:14:02.306  INFO 14496 --- [           main] org.apache.catalina.core.Stan
2023-04-13 14:14:02.660  INFO 14496 --- [           main] org.apache.jasper.servlet.Tld
2023-04-13 14:14:02.677  INFO 14496 --- [           main] o.a.c.c.C.[Tomcat].[localhost
2023-04-13 14:14:02.678  INFO 14496 --- [           main] w.s.c.ServletWebServerApplica
2023-04-13 14:14:03.150  INFO 14496 --- [           main] o.s.s.web.DefaultSecurityFilt
2023-04-13 14:14:03.607  INFO 14496 --- [           main] o.s.b.w.embedded.tomcat.Tomca
2023-04-13 14:14:03.628  INFO 14496 --- [           main] c.edu.springboot.B06SecurityA
{bcrypt}$2a$10$x4kA1EFGn35FqtG0hJLnmO9zzP4YqaDE23wGmtJkgD5F2hb8CeXtG
```

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

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://www.springframework.org/security/tags" prefix="s" %>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Welcome</title>
</head>
<body>
    <h2>Admin영역</h2>
    ADMIN권한만 접근할 수 있습니다. <br />

    <s:authorize access="hasRole('ADMIN')">
        로그인 아이디 : <s:authentication property="name"/>
    </s:authorize>

    <%@ include file="/link.jsp" %>
</body>
</html>
```

뷰 : webapp/WEB-INF/views/member.jsp

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://www.springframework.org/security/tags" prefix="s" %>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Welcome</title>
</head>
<body>
    <h2>Member영역</h2>
    USER or ADMIN권한이 있어야 접근할 수 있습니다.<br />

    <s:authorize access="isAuthenticated()">
        로그인 아이디 : <s:authentication property="name"/><br />
    </s:authorize>

    <s:authorize access="hasRole('ADMIN')">
        당신은 관리자입니다.<br />
    </s:authorize>
    <s:authorize access="hasRole('USER')">
        당신은 유저입니다.<br />
    </s:authorize>

    <%@ include file="/link.jsp" %>
</body>
</html>
```

프로젝트 설정 : resources/application.properties

```
# 포트 설정
server.port=8586

# JSP 설정
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp

# oracle 설정
spring.datasource.driver-class-name=oracle.jdbc.OracleDriver
spring.datasource.url=jdbc:oracle:thin:@localhost:1521:xe
spring.datasource.username=musthave
spring.datasource.password=1234

```

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

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

```
    @Autowired
    private DataSource dataSource;

    @Autowired
    protected void configure(AuthenticationManagerBuilder auth)
            throws Exception {
        auth.jdbcAuthentication()
            .dataSource(dataSource)
            .usersByUsernameQuery("select user_id, user_pw, enabled "
                    + " from security_admin where user_id = ?")
            .authoritiesByUsernameQuery("select user_id, authority "
                    + " from security_admin where user_id = ?")
            .passwordEncoder(new BCryptPasswordEncoder());
    }
}
```