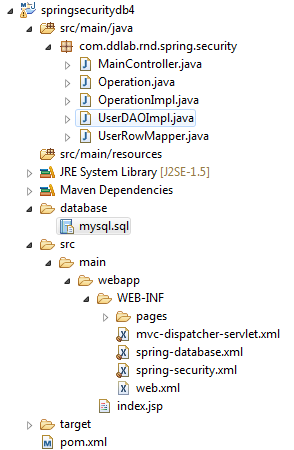
Spring Security with custom roles in web application



# Maven pom.xml

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>springsecuritydb4</groupId>

<artifactId>springsecuritydb4</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>springsecuritydb4 Maven Webapp</name>

<url>http://maven.apache.org</url>

<properties>

<spring.version>3.2.8.RELEASE</spring.version>

<spring.security.version>3.2.3.RELEASE</spring.security.version>

<jstl.version>1.2</jstl.version>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

<!-- Spring Security -->

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-web</artifactId>

<version>${spring.security.version}</version>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-config</artifactId>

<version>${spring.security.version}</version>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-taglibs</artifactId>

<version>${spring.security.version}</version>

</dependency>

<!-- jstl for jsp page -->

<dependency>

<groupId>jstl</groupId>

<artifactId>jstl</artifactId>

<version>${jstl.version}</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>servlet-api</artifactId>

<version>2.5</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>javax.servlet.jsp</groupId>

<artifactId>jsp-api</artifactId>

<version>2.1</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.30</version>

</dependency>

</dependencies>

<build>

<finalName>springsecuritydb4</finalName>

</build>

</project>

# Database SQL File

/\*Drop Table if exists\*/

**drop** **table** IF **EXISTS** appusers1 ;

/\*Create the table appusers\*/

**create** **table** appusers1

(

id **INT** AUTO\_INCREMENT **NOT** **NULL** **PRIMARY** **KEY**,

username **varchar**(30),

password **varchar**(50),

enabled BOOL,

accountNonExpired BOOL,

credentialsNonExpired BOOL,

accountNonLocked BOOL,

authority **varchar**(50),

**UNIQUE** (username)

);

/\*Insert the data into the table appusers\*/

**insert** **into** appusers1(username,password,enabled,accountNonExpired,credentialsNonExpired,accountNonLocked,authority)

**values**('deb','deb',**true**,**true**,**true**,**true**,'APP\_USER');

**commit**;

**insert** **into** appusers1(username,password,enabled,accountNonExpired,credentialsNonExpired,accountNonLocked,authority)

**values**('deb1','deb1',**true**,**true**,**true**,**true**,'APP\_WH\_USER');

**commit**;

**insert** **into** appusers1(username,password,enabled,accountNonExpired,credentialsNonExpired,accountNonLocked,authority)

**values**('deb3','deb3',**true**,**true**,**true**,**true**,'APP\_ADMIN');

**commit**;

/\*Check the data\*/

**select** \* **from** appusers1;

# Java Files

## MainController.java

package com.ddlab.rnd.spring.security;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.security.authentication.AnonymousAuthenticationToken;

import org.springframework.security.core.Authentication;

import org.springframework.security.core.context.SecurityContextHolder;

import org.springframework.security.core.userdetails.UserDetails;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.servlet.ModelAndView;

@Controller

public class MainController {

@Autowired

private Operation operation;

@RequestMapping(value = { "/", "/welcome\*\*" }, method = RequestMethod.GET)

public ModelAndView defaultPage() {

ModelAndView model = new ModelAndView();

model.addObject("title", "Spring Security Login Form - Database Authentication");

model.addObject("message", "This is default page!");

model.setViewName("hello");

return model;

}

@RequestMapping(value = "/admin\*\*", method = RequestMethod.GET)

public ModelAndView adminPage() {

ModelAndView model = new ModelAndView();

model.addObject("title", "Spring Security Login Form - Database Authentication");

model.addObject("message", "This page is for ROLE\_ADMIN only!");

model.setViewName("admin");

return model;

}

@RequestMapping(value = "/login", method = RequestMethod.GET)

public ModelAndView login(@RequestParam(value = "error", required = false) String error,

@RequestParam(value = "logout", required = false) String logout) {

ModelAndView model = new ModelAndView();

if (error != null) {

model.addObject("error", "Invalid username and password!");

}

if (logout != null) {

model.addObject("msg", "You've been logged out successfully.");

}

model.setViewName("login");

return model;

}

//for 403 access denied page

@RequestMapping(value = "/403", method = RequestMethod.GET)

public ModelAndView accesssDenied() {

ModelAndView model = new ModelAndView();

//check if user is login

Authentication auth = SecurityContextHolder.getContext().getAuthentication();

if (!(auth instanceof AnonymousAuthenticationToken)) {

UserDetails userDetail = (UserDetails) auth.getPrincipal();

model.addObject("username", userDetail.getUsername());

}

model.setViewName("403");

return model;

}

@RequestMapping(value = "/adminaction", method = RequestMethod.GET)

public ModelAndView perform() {

operation.execute();

ModelAndView model = new ModelAndView();

model.addObject("title", "Admin Operation Performance");

model.addObject("message", "Operation Successful");

model.setViewName("admin");

return model;

}

}

## Operation.java

package com.ddlab.rnd.spring.security;

import org.springframework.security.access.annotation.Secured;

public interface Operation {

// @PreAuthorize ("hasRole('APP\_ADMIN')")

// @Secured({"APP\_WH\_USER"})

// @Secured({"APP\_USER"})

// @PreAuthorize ("hasRole('APP\_ADMIN')")

//This method can only be executed by USER only

@Secured({"APP\_USER"})

public void execute();

}

## OperationImpl.java

package com.ddlab.rnd.spring.security;

public class OperationImpl implements Operation {

public void execute() {

System.out.println("Operation Executed Successfully");

}

}

## UserDAOImpl.java

package com.ddlab.rnd.spring.security;

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.security.core.userdetails.UserDetails;

import org.springframework.security.core.userdetails.UserDetailsService;

import org.springframework.security.core.userdetails.UsernameNotFoundException;

public class UserDAOImpl implements UserDetailsService {

private JdbcTemplate jdbcTemplate;

public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {

String sql = "SELECT \* FROM appusers1 WHERE username = ?";

UserDetails user = jdbcTemplate.queryForObject(sql, new Object[] { username }, new UserRowMapper());

return user;

}

public void setJdbcTemplate(JdbcTemplate jdbcTemplate) {

this.jdbcTemplate = jdbcTemplate;

}

}

## UserRowMapper.java

package com.ddlab.rnd.spring.security;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import org.springframework.jdbc.core.RowMapper;

import org.springframework.security.core.GrantedAuthority;

import org.springframework.security.core.authority.SimpleGrantedAuthority;

import org.springframework.security.core.userdetails.User;

public class UserRowMapper implements RowMapper<User>

{

public User mapRow(ResultSet rs, int rowNum) throws SQLException {

User user = null;

String username = rs.getString("username");

String password = rs.getString("password");

boolean enabled = rs.getBoolean("enabled");

boolean accountNonExpired = rs.getBoolean("accountNonExpired");

boolean credentialsNonExpired = rs.getBoolean("credentialsNonExpired");

boolean accountNonLocked = rs.getBoolean("accountNonLocked");

String authority = rs.getString("authority");

List<GrantedAuthority> gaList = new ArrayList<GrantedAuthority>();

SimpleGrantedAuthority auth1 = new SimpleGrantedAuthority(authority);

gaList.add(auth1);

user = new User(username, password, enabled, accountNonExpired,

credentialsNonExpired, accountNonLocked, gaList);

return user;

}

}

# WEB-Files

## Jsp files

### 403.jsp

<%@taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<html>

<body>

<h1>HTTP Status 403 - Access is denied</h1>

<c:url value=*"/j\_spring\_security\_logout"* var=*"logoutUrl"* />

<form action=*"*${logoutUrl}*"* method=*"post"* id=*"logoutForm"*>

<input type=*"hidden"* name=*"*${\_csrf.parameterName}*"*

value=*"*${\_csrf.token}*"* />

</form>

<script>

**function** formSubmit() {

document.getElementById("logoutForm").submit();

}

</script>

<c:choose>

<c:when test=*"*${**empty** username}*"*>

<h2>You do not have permission to access this page!</h2>

</c:when>

<c:otherwise>

<h2>Username : ${username} <br/>You do not have permission to access this page!</h2>

<a href=*"javascript:formSubmit()"*> Logout</a>

</c:otherwise>

</c:choose>

</body>

</html>

### Admin.jsp

<%@taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<%@page session=*"true"*%>

<html>

<body>

<h1>Title : ${title}</h1>

<h1>Message : ${message}</h1>

<c:url value=*"/j\_spring\_security\_logout"* var=*"logoutUrl"* />

<form action=*"*${logoutUrl}*"* method=*"post"* id=*"logoutForm"*>

<input type=*"hidden"* name=*"*${\_csrf.parameterName}*"*

value=*"*${\_csrf.token}*"* />

</form>

<script>

**function** formSubmit() {

document.getElementById("logoutForm").submit();

}

</script>

<c:if test=*"*${pageContext.request.userPrincipal.name != null}*"*>

<h2>

Welcome : ${pageContext.request.userPrincipal.name} | <a

href=*"javascript:formSubmit()"*> Logout</a>

</h2>

</c:if>

<form action=*"adminaction"* method=*"GET"* id=*"adminactionId"*></form>

<script>

**function** perform() {

document.getElementById("adminactionId").submit();

}

</script>

<!-- It will be executed only by Admin -->

<a href=*"javascript:perform()"*> Perform Operation</a>

</body>

</html>

### Hello.jsp

<%@taglib prefix=*"sec"*

uri=*"http://www.springframework.org/security/tags"*%>

<%@taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<html>

<body>

<h1>Title : ${title}</h1>

<h1>Message : ${message}</h1>

<%-- <sec:authorize access="hasRole('ROLE\_USER')"> --%>

<!-- For login user -->

<c:url value=*"/j\_spring\_security\_logout"* var=*"logoutUrl"* />

<form action=*"*${logoutUrl}*"* method=*"post"* id=*"logoutForm"*>

<input type=*"hidden"* name=*"*${\_csrf.parameterName}*"*

value=*"*${\_csrf.token}*"* />

</form>

<script>

**function** formSubmit() {

document.getElementById("logoutForm").submit();

}

</script>

<c:if test=*"*${pageContext.request.userPrincipal.name != null}*"*>

<h2>

User : ${pageContext.request.userPrincipal.name} | <a

href=*"javascript:formSubmit()"*> Logout</a>

</h2>

</c:if>

<%-- </sec:authorize> --%>

</body>

</html>

### Login.jsp

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<%@page session=*"true"*%>

<html>

<head>

<title>Login Page</title>

<style>

*.error* {

padding: *15px*;

margin-bottom: *20px*;

border: *1px solid transparent*;

border-radius: *4px*;

color: *#a94442*;

background-color: *#f2dede*;

border-color: *#ebccd1*;

}

*.msg* {

padding: *15px*;

margin-bottom: *20px*;

border: *1px solid transparent*;

border-radius: *4px*;

color: *#31708f*;

background-color: *#d9edf7*;

border-color: *#bce8f1*;

}

*#login-box* {

width: *300px*;

padding: *20px*;

margin: *100px auto*;

background: *#fff*;

-webkit-border-radius: *2px*;

-moz-border-radius: *2px*;

border: *1px solid #000*;

}

</style>

</head>

<body onload='document.loginForm.username.focus();'>

<h1>Spring Security Login Form (Database Authentication)</h1>

<div id=*"login-box"*>

<h3>Login with Username and Password</h3>

<c:if test=*"*${**not empty** error}*"*>

<div class=*"error"*>${error}</div>

</c:if>

<c:if test=*"*${**not empty** msg}*"*>

<div class=*"msg"*>${msg}</div>

</c:if>

<form name=*'loginForm'*

action=*"*<c:url value=*'/j\_spring\_security\_check'* />*"* method=*'POST'*>

<table>

<tr>

<td>User:</td>

<td><input type=*'text'* name=*'username'*></td>

</tr>

<tr>

<td>Password:</td>

<td><input type=*'password'* name=*'password'* /></td>

</tr>

<tr>

<td colspan=*'2'*><input name=*"submit"* type=*"submit"*

value=*"submit"* /></td>

</tr>

</table>

<input type=*"hidden"* name=*"*${\_csrf.parameterName}*"*

value=*"*${\_csrf.token}*"* />

</form>

</div>

</body>

</html>

### Index.jsp

<html>

<body>

<h2>Hello World!</h2>

</body>

</html>

## WEB Config files

### Mvc-dispatcher-servlet.xml

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.0.xsd"*>

**<context:component-scan base-package=*"com.ddlab.rnd.spring.\*"* />**

<bean

class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"prefix"*>

<value>/WEB-INF/pages/</value>

</property>

<property name=*"suffix"*>

<value>.jsp</value>

</property>

</bean>

</beans>

### Spring-database.xml

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd"*>

<bean id=*"dataSource"*

class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"* value=*"com.mysql.jdbc.Driver"* />

<property name=*"url"* value=*"jdbc:mysql://localhost:3306/test"* />

<property name=*"username"* value=*"root"* />

<property name=*"password"* value=*"root"* />

</bean>

</beans>

### Spring-security.xml

<beans:beans xmlns=*"http://www.springframework.org/schema/security"*

xmlns:beans=*"http://www.springframework.org/schema/beans"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/security*

*http://www.springframework.org/schema/security/spring-security-3.2.xsd"*>

**<beans:bean id=*"roleHierarchy"* class=*"org.springframework.security.access.hierarchicalroles.RoleHierarchyImpl"*>**

**<beans:property name=*"hierarchy"*>**

**<beans:value>**

**APP\_ADMIN > APP\_WH\_USER**

**APP\_WH\_USER > APP\_USER**

**</beans:value>**

**</beans:property>**

**</beans:bean>**

**<beans:bean id=*"roleVoter"* class=*"org.springframework.security.access.vote.RoleHierarchyVoter"*>**

**<beans:constructor-arg>**

**<beans:ref bean=*"roleHierarchy"* />**

**</beans:constructor-arg>**

**</beans:bean>**

**<beans:bean id=*"expressionHandler"* class=*"org.springframework.security.web.access.expression.DefaultWebSecurityExpressionHandler"*>**

**<beans:property name=*"roleHierarchy"* ref=*"roleHierarchy"* />**

**</beans:bean>**

**<beans:bean id=*"webExpressionHandler"* class=*"org.springframework.security.web.access.expression.WebExpressionVoter"*>**

**<beans:property name=*"expressionHandler"*>**

**<beans:ref bean=*"expressionHandler"* />**

**</beans:property>**

**</beans:bean>**

**<beans:bean id=*"accessDecisionManager"* class=*"org.springframework.security.access.vote.AffirmativeBased"*>**

**<beans:constructor-arg>**

**<beans:list>**

**<beans:ref bean=*"roleVoter"* />**

**<beans:bean class=*"org.springframework.security.access.vote.AuthenticatedVoter"* />**

**<beans:ref bean=*"webExpressionHandler"* />**

**</beans:list>**

**</beans:constructor-arg>**

**</beans:bean>**

**<beans:bean id=*"defaultMethodSecurityExpressionHandler"***

**class=*"org.springframework.security.access.expression.method.DefaultMethodSecurityExpressionHandler"*>**

**<beans:property name=*"roleHierarchy"* ref=*"roleHierarchy"* />**

**</beans:bean>**

**<global-method-security secured-annotations=*"enabled"* pre-post-annotations=*"enabled"*>**

**<expression-handler ref=*"defaultMethodSecurityExpressionHandler"* />**

**</global-method-security>**

<!-- enable use-expressions -->

<http auto-config=*"true"* use-expressions=*"true"*>

<intercept-url pattern=*"/admin\*\*"* access=*"hasRole('APP\_ADMIN')"* />

<!-- access denied page -->

<access-denied-handler error-page=*"/403"* />

<form-login login-page=*"/login"* default-target-url=*"/welcome"*

authentication-failure-url=*"/login?error"* username-parameter=*"username"*

password-parameter=*"password"* />

<logout logout-success-url=*"/login?logout"* />

<!-- enable csrf protection -->

<csrf />

</http>

**<beans:bean id=*"userDAO"* class=*"com.ddlab.rnd.spring.security.UserDAOImpl"*>**

**<beans:property name=*"jdbcTemplate"* ref=*"jdbcTemplate"*></beans:property>**

**</beans:bean>**

<!-- Select users and user\_roles from database -->

**<authentication-manager>**

**<authentication-provider user-service-ref=*"userDAO"*>**

**</authentication-provider>**

**</authentication-manager>**

<beans:bean id=*"jdbcTemplate"* class=*"org.springframework.jdbc.core.JdbcTemplate"*>

<beans:property name=*"dataSource"* ref=*"dataSource"*></beans:property>

</beans:bean>

<beans:bean id=*"operation"* class=*"com.ddlab.rnd.spring.security.OperationImpl"*/>

</beans:beans>

### Web.xml

<web-app id=*"WebApp\_ID"* version=*"2.4"*

xmlns=*"http://java.sun.com/xml/ns/j2ee"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://java.sun.com/xml/ns/j2ee*

*http://java.sun.com/xml/ns/j2ee/web-app\_2\_4.xsd"*>

<display-name>Spring Security Application</display-name>

<!-- Spring MVC -->

<servlet>

<servlet-name>mvc-dispatcher</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>mvc-dispatcher</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>

/WEB-INF/spring-security.xml,

/WEB-INF/spring-database.xml

</param-value>

</context-param>

<!-- Spring Security -->

<filter>

<filter-name>springSecurityFilterChain</filter-name>

<filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>

</filter>

<filter-mapping>

<filter-name>springSecurityFilterChain</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

</web-app>