SpringSecurity入门版本

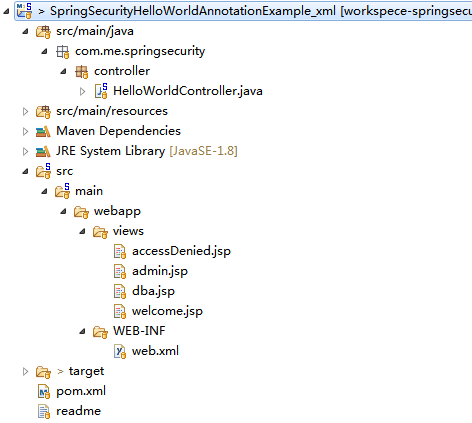
xml配置版本

这个项目是根据网上的例子来照着做的。使用的SpringMVC整合SpringSecurity4的例子。

参考资料：<http://websystique.com/>

基于web.xml配置的springmv + springSecurity4的例子。

先上一个项目的结构图



## 首先

使用maven建立一个web项目（虽然我们这里使用的是java配置，没有用到web.xml，但后面的页面还是放在webapp这一web层中），并命名为 SpringSecurityHelloWorldAnnotationExample。

然后再pom.xml中引入下面的jar包（也可以全部复制过去）

<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/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.me</groupId>

<artifactId>SpringSecurityHelloWorldAnnotationExample\_xml</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>war</packaging>

<name>SpringSecurityHelloWorldAnnotationExample\_xml</name>

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

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<springframework.version>4.3.8.RELEASE</springframework.version>

<springsecurity.version>4.0.1.RELEASE</springsecurity.version>

</properties>

<dependencies>

<!-- Spring -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

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

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

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

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

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

</dependency>

<!-- Spring Security -->

<dependency>

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

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

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

</dependency>

<dependency>

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

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

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

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>3.1.0</version>

</dependency>

<dependency>

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

<artifactId>javax.servlet.jsp-api</artifactId>

<version>2.3.1</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

</dependencies>

<build>

<pluginManagement>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.2</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

<!--

//这个模块是用来将项目点打包的时候避免不配置web.xml时报错。用于不用配置web.xml时使用

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-war-plugin</artifactId>

<version>2.4</version>

<configuration>

<warSourceDirectory>src/main/webapp</warSourceDirectory>

<warName>SpringSecurityHelloWorldAnnotationExample\_xml</warName>

<failOnMissingWebXml>false</failOnMissingWebXml>

</configuration>

</plugin> -->

</plugins>

</pluginManagement>

<finalName>SpringSecurityHelloWorldAnnotationExample\_xml</finalName>

</build>

</project>

#### **第一步：**创建web.xml文件

主要是引入springmvc 和springSecurity的加载类。

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

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

xmlns=*"http://java.sun.com/xml/ns/javaee"*

xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"*

id=*"WebApp\_ID"* version=*"3.0"*>

<filter>

<filter-name>CharacterEncodingFilter</filter-name>

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

<init-param>

<param-name>encoding</param-name>

<param-value>utf-8</param-value>

</init-param>

<init-param>

<param-name>forceEncoding</param-name>

<param-value>true</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>CharacterEncodingFilter</filter-name>

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

</filter-mapping>

<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>

<servlet>

<servlet-name>SpringMVC</servlet-name>

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

<init-param>

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

<param-value>classpath:spring/spring-mvc.xml</param-value>

</init-param>

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

</servlet>

<servlet-mapping>

<servlet-name>SpringMVC</servlet-name>

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

</servlet-mapping>

<welcome-file-list>

<welcome-file>/</welcome-file>

</welcome-file-list>

</web-app>

#### **第二步：**编写springmvc配置文件。

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

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

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

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

xsi:schemaLocation=*"http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.3.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd*

*http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.3.xsd"*>

<!-- 配置扫描controller的所有注解，是spring的容器 -->

<context:component-scan base-package=*"com.me.\*\*"*>

<context:include-filter type=*"annotation"* expression=*"org.springframework.stereotype.Controller"* />

</context:component-scan>

<!-- 使用自动注册方法,这个是springmvc容器的扫描注解，配置requestmapping使用，没有这个，会报找不到Controller -->

<mvc:annotation-driven />

<!-- 配置jsp显示的ViewResolver -->

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

<property name=*"viewClass"*

value=*"org.springframework.web.servlet.view.JstlView"* />

<!-- 前缀 -->

<property name=*"prefix"* value=*"/views/"* />

<!-- 后缀 -->

<property name=*"suffix"* value=*".jsp"* />

</bean>

<import resource=*"spring-security.xml"*/>

</beans>

**第三步：编写springSecurity配置文件**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

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

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

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

xsi:schemaLocation=*"http://www.springframework.org/schema/security http://www.springframework.org/schema/security/spring-security-4.0.xsd*

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

<s:http auto-config=*"true"*>

<s:intercept-url pattern=*"/"* access=*"permitAll"*/>

<s:intercept-url pattern=*"/home"* access=*"permitAll"*/>

<s:intercept-url pattern=*"/admin\*\*"* access=*"hasRole('ADMIN')"*/>

<s:intercept-url pattern=*"/db\*\*"* access=*"hasRole('ADMIN') and hasRole('DBA')"*/>

<s:form-login authentication-failure-url=*"/Access\_Denied"*/>

</s:http>

<s:authentication-manager>

<s:authentication-provider>

<s:user-service>

<s:user name=*"bill"* password=*"bill"* authorities=*"ROLE\_USER"*/>

<s:user name=*"admin"* password=*"admin"* authorities=*"ROLE\_ADMIN"*/>

<s:user name=*"dba"* password=*"dba"* authorities=*"ROLE\_ADMIN,ROLE\_DBA"*/>

</s:user-service>

</s:authentication-provider>

</s:authentication-manager>

</beans>

#### 第四步：添加 Controller（控制器）

com.websystique.springsecurity.controller.HelloWorldController

package com.websystique.springsecurity.controller;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.security.core.Authentication;

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

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

import org.springframework.security.web.authentication.logout.SecurityContextLogoutHandler;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

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

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

@Controller

public class HelloWorldController {

@RequestMapping(value = { "/", "/home" }, method = RequestMethod.GET)

public String homePage(ModelMap model) {

model.addAttribute("greeting", "Hi, Welcome to mysite. ");

return "welcome";

}

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

public String adminPage(ModelMap model) {

model.addAttribute("user", getPrincipal());

return "admin";

}

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

public String dbaPage(ModelMap model) {

model.addAttribute("user", getPrincipal());

return "dba";

}

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

public String logoutPage (HttpServletRequest request, HttpServletResponse response) {

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

if (auth != null){

new SecurityContextLogoutHandler().logout(request, response, auth);

}

return "welcome";

}

@RequestMapping(value = "/Access\_Denied", method = RequestMethod.GET)

public String accessDeniedPage(ModelMap model) {

model.addAttribute("user", getPrincipal());

return "accessDenied";

}

private String getPrincipal(){

String userName = null;

Object principal = SecurityContextHolder.getContext().getAuthentication().getPrincipal();

if (principal instanceof UserDetails) {

userName = ((UserDetails)principal).getUsername();

} else {

userName = principal.toString();

}

return userName;

}

}

controller类中的方法比较繁琐.getPrincipal 方法返回从Spring SecurityContext中记录的登录的用户。

logoutPage 方法简单调用 SecurityContextLogoutHandler().logout(request, response, auth)方法  
来处理退出操作。

它很巧妙而且将你从不容易管理的jsp页面退出逻辑中解放出来。

你也许注意到上面没有出现 /login’，因为Spring Security默认会产生和处理。

#### 第五步: 添加Views（视图）

login.jsp

(该页面中加入了几个css文件，主要是为了显示效果。如果不需要可以不引入。不影响实际功能)

<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>

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

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Login page</title>

<link href="<c:url value='/static/css/bootstrap.css' />" rel="stylesheet"></link>

<link href="<c:url value='/static/css/app.css' />" rel="stylesheet"></link>

<link rel="stylesheet" type="text/css" href="//cdnjs.cloudflare.com/ajax/libs/font-awesome/4.2.0/css/font-awesome.css" />

</head>

<body>

<div id="mainWrapper">

<div class="login-container">

<div class="login-card">

<div class="login-form">

<c:url var="loginUrl" value="/login" />

<form action="${loginUrl}" method="post" class="form-horizontal">

<c:if test="${param.error != null}">

<div class="alert alert-danger">

<p>Invalid username and password.</p>

</div>

</c:if>

<c:if test="${param.logout != null}">

<div class="alert alert-success">

<p>You have been logged out successfully.</p>

</div>

</c:if>

<div class="input-group input-sm">

<label class="input-group-addon" for="username"><i class="fa fa-user"></i></label>

<input type="text" class="form-control" id="username" name="ssoId" placeholder="Enter Username" required>

</div>

<div class="input-group input-sm">

<label class="input-group-addon" for="password"><i class="fa fa-lock"></i></label>

<input type="password" class="form-control" id="password" name="password" placeholder="Enter Password" required>

</div>

<input type="hidden" name="${\_csrf.parameterName}" value="${\_csrf.token}" />

<div class="form-actions">

<input type="submit"

class="btn btn-block btn-primary btn-default" value="Log in">

</div>

</form>

</div>

</div>

</div>

</div>

</body>

</html>

welcome.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>HelloWorld page</title>

</head>

<body>

Greeting : ${greeting}

This is a welcome page.

</body>

</html>

admin.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>

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

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>HelloWorld Admin page</title>

</head>

<body>

Dear <strong>${user}</strong>, Welcome to Admin Page.

<a href="<c:url value="/logout" />">Logout</a>

</body>

</html>

dba.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>

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

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>DBA page</title>

</head>

<body>

Dear <strong>${user}</strong>, Welcome to DBA Page.

<a href="<c:url value="/logout" />">Logout</a>

</body>

</html>

accessDenied.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>

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

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>AccessDenied page</title>

</head>

<body>

Dear <strong>${user}</strong>, You are not authorized to access this page

<a href="<c:url value="/logout" />">Logout</a>

</body>

</html>

#### 第六步: 发布应用

使用tomcat（可以更换其他的容器）启动应用  
打开浏览器 在地址栏输入 [localhost:8080/SpringSecurityHelloWorldAnnotationExample\_xml/](localhost:8080/SpringSecurityHelloWorldAnnotationExample_xml/admin)并回车就能看到页面。页面和上一个java配置版本一样。这里就不演示了。

参考资料源码下载地址：<http://websystique.com/?smd_process_download=1&amp;download_id=1334>