spring-security与springmvc整合

原创x286129277 最后发布于2018-11-03 18:54:30 阅读数 2914 收藏

展开

一、导入pom坐标

<dependency>

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

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

<version>5.0.1.RELEASE</version>

</dependency>

<dependency>

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

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

<version>5.0.1.RELEASE</version>

</dependency>

<dependency>

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

<artifactId>spring-security-core</artifactId>

<version>5.0.1.RELEASE</version>

</dependency>

<dependency>

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

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

<version>5.0.1.RELEASE</version>

</dependency>

二、配置web.xml

注意：配置spring-security的过滤器时，filter-name的名称不能随意修改，只能为配置中所写的名称。

<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>

三、配置spring-security.xml配置信息

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

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

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

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.xsd

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

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

<!-- 配置不拦截的资源 -->

<security:http pattern="/login.jsp" security="none"/>

<security:http pattern="/failer.jsp" security="none"/>

<security:http pattern="/css/\*\*" security="none"/>

<security:http pattern="/img/\*\*" security="none"/>

<security:http pattern="/plugins/\*\*" security="none"/>

<!--

配置具体的规则

auto-config="true" 不用自己编写登录的页面，框架提供默认登录页面

use-expressions="false" 是否使用SPEL表达式

-->

<security:http auto-config="true" use-expressions="false">

<!-- 配置具体的拦截的规则 pattern="请求路径的规则" access="访问系统的人，必须有ROLE\_USER的角色" -->

<security:intercept-url pattern="/\*\*" access="ROLE\_USER,ROLE\_ADMIN"/>

<!-- 定义跳转的具体的页面 -->

<security:form-login

login-page="/login.jsp"

login-processing-url="/login.do"

default-target-url="/index.jsp"

authentication-failure-url="/failer.jsp"

/>

<!-- 关闭跨域请求 -->

<security:csrf disabled="true"/>

<!-- 退出 -->

<security:logout invalidate-session="true" logout-url="/logout.do" logout-success-url="/login.jsp" />

</security:http>

<!-- 切换成数据库中的用户名和密码 -->

<security:authentication-manager>

<security:authentication-provider user-service-ref="userService">

<!-- 配置加密的方式 -->

<security:password-encoder ref="passwordEncoder"/>

</security:authentication-provider>

</security:authentication-manager>

<!-- 配置加密类 -->

<bean id="passwordEncoder" class="org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder"/>

</beans>

四、配置自定义UserDetailService实现类

注意：该自定义的service中，@Service中的注解名称，必须与Spring-security.xml中指定的<security:authentication-provider user-service-ref="userService">名称必须相同。

@Service("userService")

public class UserServiceImpl implements UserDetailsService {

@Override

public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {

//第一步：通过用户名，去数据库查询用户信息

//第二步：根据用户信息，实例化，处理自己的用户对象封装成UserDetails

//如果是使用未加密的帐号和密码校验，请在用户名和密码前面，添加一个标识{noop}

// User user=new User(userInfo.getUsername(),"{noop}"+userInfo.getPassword(),getAuthority(userInfo.getRoles()));

User user = new User(登录用户名, 从数据库查询出来的用户密码, 用户状态，false表示无效，true表示有效, true, true, true, 获取当前登录用户的角色列表));

return user;

}

//作用就是返回一个List集合，集合中装入的是角色描述

public List<SimpleGrantedAuthority> getAuthority(List<Role> roles) {

List<SimpleGrantedAuthority> list = new ArrayList<>();

for (Role role : roles) {

list.add(new SimpleGrantedAuthority("ROLE\_" + role.getRoleName()));

}

return list;

}

}

五、测试

1、重新登录系统，访问任何一个请求，如果直接都跳转到 login.jsp的登录界面，则表示框架整合成功

2、输入一个正确的帐号和密码，如果报403（权限不足），则表示当前用户配置的角色权限有问题，请检查自定义实现的UserDetailServiceImpl类中，获取的用户角色列表名称中，是否有spring-security.xml中<security:intercept-url pattern="/\*\*" access="ROLE\_USER,ROLE\_ADMIN"/> 任何一个角色。

————————————————

版权声明：本文为CSDN博主「x286129277」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/x286129277/java/article/details/83689409