Shiro学习笔记—spring集成

|  |  |
| --- | --- |
|  | 🢂 内容概览 |
|  | Why：此文档用来做什么？它存在的意义是什么？为解决什么问题？   |  | | --- | |  |   What：当前包含了那些内容？   |  | | --- | |  |   How：此文档应如何参考？   |  | | --- | |  |   Who：此文档适用于那些人员阅读参考？   |  | | --- | |  | |

目录

找不到标题。 **这是一个自动目录。若要使用它，请对进入目录的文本应用标题样式(“开始”选项卡上)，然后更新此目录。 如果您想要键入自己的条目，请使用手动目录(与手动目录位于相同菜单中)。**

# Spring集成

## 配置maven依赖

|  |
| --- |
| *<!-- shiro -->* <**dependency**>  <**groupId**>org.apache.shiro</**groupId**>  <**artifactId**>shiro-core</**artifactId**>  <**version**>${shiro.core.version}</**version**> </**dependency**> <**dependency**>  <**groupId**>org.apache.shiro</**groupId**>  <**artifactId**>shiro-spring</**artifactId**>  <**version**>${shiro.core.version}</**version**> </**dependency**> <**dependency**>  <**groupId**>org.apache.shiro</**groupId**>  <**artifactId**>shiro-ehcache</**artifactId**>  <**version**>${shiro.core.version}</**version**> </**dependency**> <**dependency**>  <**groupId**>org.apache.shiro</**groupId**>  <**artifactId**>shiro-web</**artifactId**>  <**version**>${shiro.core.version}</**version**> </**dependency**> |

## 配置shiro过滤器

在web.xml中配置shiro过滤器，用来过滤所有需要shiro处理的请求；该过滤器应放在所有过滤器最前面，以保证在访问其它资源之前时，先经过shiro过滤；

|  |
| --- |
|  |

## 配置过滤器bean

在spring配置文件中定义过滤器bean，名字与web.xml中过滤器名字相同；

|  |
| --- |
| <bean id="shiroFilter" class="org.apache.shiro.spring.web.ShiroFilterFactoryBean">  <property name="securityManager" ref="securityManager"/>  <!-- override these for application-specific URLs if you like:  <property name="loginUrl" value="/login.jsp"/>  <property name="successUrl" value="/home.jsp"/>  <property name="unauthorizedUrl" value="/unauthorized.jsp"/> -->  <!-- The 'filters' property is not necessary since any declared javax.servlet.Filter bean -->  <!-- defined will be automatically acquired and available via its beanName in chain -->  <!-- definitions, but you can perform instance overrides or name aliases here if you like: -->  <!-- <property name="filters">  <util:map>  <entry key="anAlias" value-ref="someFilter"/>  </util:map>  </property> -->  <property name="filterChainDefinitions">  <value>  # some example chain definitions:  /admin/\*\* = authc, roles[admin]  /docs/\*\* = authc, perms[document:read]  /\*\* = authc  # more URL-to-FilterChain definitions here  </value>  </property>  </bean>  <!-- Define any javax.servlet.Filter beans you want anywhere in this application context. -->  <!-- They will automatically be acquired by the 'shiroFilter' bean above and made available -->  <!-- to the 'filterChainDefinitions' property. Or you can manually/explicitly add them -->  <!-- to the shiroFilter's 'filters' Map if desired. See its JavaDoc for more details. -->  <bean id="someFilter" class="..."/>  <bean id="anotherFilter" class="..."> ... </bean>  ...  <bean id="securityManager" class="org.apache.shiro.web.mgt.DefaultWebSecurityManager">  <!-- Single realm app. If you have multiple realms, use the 'realms' property instead. -->  <property name="realm" ref="myRealm"/>  <!-- By default the servlet container sessions will be used. Uncomment this line  to use shiro's native sessions (see the JavaDoc for more): -->  <!-- <property name="sessionMode" value="native"/> -->  </bean>  <bean id="lifecycleBeanPostProcessor" class="org.apache.shiro.spring.LifecycleBeanPostProcessor"/>  <!-- Define the Shiro Realm implementation you want to use to connect to your back-end -->  <!-- security datasource: -->  <bean id="myRealm" class="...">  ...  </bean> |

## 启用shiro注解

|  |
| --- |
| <!-- Enable Shiro Annotations for Spring-configured beans. Only run after -->  <!-- the lifecycleBeanProcessor has run: -->  <bean class="org.springframework.aop.framework.autoproxy.DefaultAdvisorAutoProxyCreator" depends-on="lifecycleBeanPostProcessor"/>  <bean class="org.apache.shiro.spring.security.interceptor.AuthorizationAttributeSourceAdvisor">  <property name="securityManager" ref="securityManager"/>  </bean> |