Shiro学习笔记

# Shiro基本配置

1. 导入jar包：

log4j-1.2.15.jar

shiro-all-1.3.2.jar

slf4j-api-1.6.1.jar

slf4j-log4j12-1.6.1.jar

1. 复制\shiro-root-1.2.2-source-release\shiro-root-1.2.2\samples\quickstart\src\main\resources路径下的文件



1. 使用Shiro，执行Quickstart.java的main方法。

# Web下使用Shiro（集成Spring+SpringMVC）

1. Web下配置Spring+SpringMVC
   1. 导入jar包
   2. web.xml下配置Spring+SpringMVC
   3. 分别配置applicationContext.xml和springmvc.xml文件
2. 在web.xml文件中配置Shiro

|  |
| --- |
| <filter>  <filter-name>shiroFilter</filter-name>  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>  <init-param>  <param-name>targetFilterLifecycle</param-name>  <param-value>true</param-value>  </init-param>  </filter>  <filter-mapping>  <filter-name>shiroFilter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping> |

1. 在applicationContext.xml中配置Shiro

|  |
| --- |
| <!-- 配置securityManager-->  <bean id="securityManager" class="org.apache.shiro.web.mgt.DefaultWebSecurityManager">  <property name="cacheManager" ref="cacheManager"/>  <property name="realm" ref="jdbcRealm"/>  </bean>  <!-- 配置CacheManager,需要加入ehcache 的jar包及ehcache.xml文件.-->  <bean id="cacheManager" class="org.apache.shiro.cache.ehcache.EhCacheManager">  <property name="cacheManagerConfigFile" value="classpath:ehcache.xml"/>  </bean>  <!-- 配置 Realm，自定义实现 org.apache.shiro.realm.Realm 接口的 类 -->  <bean id="jdbcRealm" class="spring.ShiroRealm"></bean>  <bean id="lifecycleBeanPostProcessor" class="org.apache.shiro.spring.LifecycleBeanPostProcessor"/>  <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>  <bean id="shiroFilter" class="org.apache.shiro.spring.web.ShiroFilterFactoryBean">  <property name="securityManager" ref="securityManager"/>  <property name="loginUrl" value="/login.jsp"/>  <property name="successUrl" value="/list.jsp"/>  <property name="unauthorizedUrl" value="/unauthorized.jsp"/>  <property name="filterChainDefinitions">  <value>  /login.jsp = anon  /\*\* = authc  </value>  </property>  </bean> |

# 实现Shiro认证

1. 自定义类，继承AuthenticatingRealm，重写doGetAuthenticationInfo方法

|  |
| --- |
| UsernamePasswordToken upt=(UsernamePasswordToken) token;  String name=upt.getUsername();  String credentials="123456";  if("unknow".equals(name))  {  throw new UnknownAccountException(name+" 用户不存在");  }else if("root".equals(name)) {  throw new LockedAccountException(name+"用户被锁定");  }else if("user".equals(name)){  credentials = "b8c2d5b0a37cc51f91d5e8970347a3a3";  }else if("admin".equals(name)){  credentials = "928bfd2577490322a6e19b793691467e";  }  // SimpleAuthenticationInfo info=new SimpleAuthenticationInfo(name,"123456",getName());    ByteSource salt=ByteSource.Util.bytes(name);  SimpleAuthenticationInfo info=new SimpleAuthenticationInfo(name,credentials,salt,getName());  return info; |

1. applicationContext.xml中配置认证：

|  |
| --- |
| <bean id="securityManager" class="org.apache.shiro.web.mgt.DefaultWebSecurityManager">  <property name="cacheManager" ref="cacheManager"/>  <property name="authenticator" ref="authenticator"/>  <property name="realms">  <list>  <ref bean="jdbcRealm"/>  <ref bean="shiroRealm2"/>  </list>  </property>  </bean> |
| <bean id=*"authenticator"* class=*"org.apache.shiro.authc.pam.ModularRealmAuthenticator"*>    <property name=*"authenticationStrategy"*>  <bean class=*"org.apache.shiro.authc.pam.AtLeastOneSuccessfulStrategy"*/>  </property>  </bean>    <bean id=*"jdbcRealm"* class=*"shiro.ShiroRealm"*>  <property name=*"credentialsMatcher"*>  <bean class=*"org.apache.shiro.authc.credential.HashedCredentialsMatcher"*>  <property name=*"hashAlgorithmName"* value=*"MD5"*></property>  <property name=*"hashIterations"* value=*"2"*></property>  </bean>  </property>  </bean>  <bean id=*"shiroRealm2"* class=*"shiro.ShiroRealm2"*>  <property name=*"credentialsMatcher"*>  <bean class=*"org.apache.shiro.authc.credential.HashedCredentialsMatcher"*>  <property name=*"hashAlgorithmName"* value=*"SHA1"*></property>  <property name=*"hashIterations"* value=*"2"*></property>  </bean>  </property>  </bean> |

# 实现Shiro授权

1. 自定义类，继承AuthorizingRealm，重写doGetAuthorizationInfo方法

|  |
| --- |
| protected AuthorizationInfo doGetAuthorizationInfo(PrincipalCollection principals) {  System.out.println("正在授权。。。");  Object principal = principals.getPrimaryPrincipal();  Set<String> roles=new HashSet<String>();  roles.add("user");  if("admin".equals(principal))  {  roles.add("admin");  }  AuthorizationInfo info=new SimpleAuthorizationInfo(roles);  return info;  } |

# Shiro标签

1. 常见标签及用法：

# Shiro注解

1. 权限注解：