# Shiro框架开发步骤

使用shiro框架进行项目的认证和授权操作

### 第一步：导入shiro的jar到项目中

### 第二步：在web.xml中配置一个过滤器代理对象，在项目启动时到spring工厂中加载一个和当前过滤器name同名的bean对象

web.xml

<!-- 加入spring框架提供的过滤器代理对象 -->

<filter>

<filter-name>shiroFilter</filter-name>

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

</filter>

<filter-mapping>

<filter-name>shiroFilter</filter-name>

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

</filter-mapping>

### 第三步：在spring配置文件中配置一个名称为shiroFilter的bean

<!-- 配置shiro的bean， 使用当前工厂bean对象创建过滤器用于进行权限控制-->

<bean id="shiroFilter" class="org.apache.shiro.spring.web.ShiroFilterFactoryBean">

<!-- 注入安全管理器对象 -->

<property name="securityManager" ref="securityManager"></property>

<!-- 登录页面url地址 -->

<property name="loginUrl" value="/login.jsp"></property>

<!-- 成功页面 -->

<property name="successUrl" value="/index.jsp"></property>

<!-- 权限不足提示页面 -->

<property name="unauthorizedUrl" value="/unauthorized.jsp"></property>

<!-- 基于url拦截，使用过滤器进行拦截 -->

<property name="filterChainDefinitions">

<value>

/css/\*\* = anon

/images/\*\* = anon

/js/\*\* = anon

/validatecode.jsp\* = anon

/login.jsp = anon

/userAction\_login.action = anon

/page\_base\_staff.action = perms["staff"]

/\*\* = authc

</value>

</property>

</bean>

<!-- 配置安全管理器 -->

<bean id="securityManager" class="org.apache.shiro.web.mgt.DefaultWebSecurityManager">

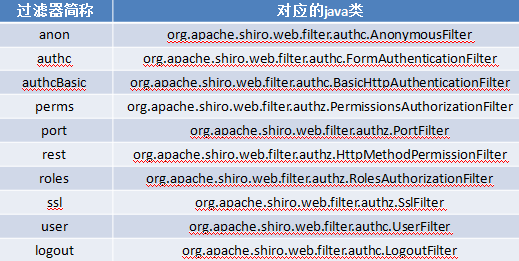
<property name="realm" ref="bosRealm"></property>

</bean>

<!-- 配置自定义的realm -->

<bean id="bosRealm" class="cn.itcast.bos.shiro.BOSRealm"/>

Shiro框架提供的过滤器：



### 第四步：修改UserAction的login登录访问，使用shiro框架提供的方式进行认证操作

//验证码输入成功

//使用shiro提供的方式进行权限认证

//获得当前用户对象，现在状态为“未认证”

Subject subject = SecurityUtils.*getSubject*();

String username = model.getUsername();

String password = model.getPassword();

password = MD5Utils.*md5*(password);

AuthenticationToken token = **new** UsernamePasswordToken(username,password);

**try**{

subject.login(token);//调用安全管理器，安全管理器调用Realm

User user = (User) subject.getPrincipal();

// 登录成功，将user放入session，跳转到系统首页

ServletActionContext.*getRequest*().getSession().setAttribute("loginUser", user);

}**catch** (UnknownAccountException e) {

e.printStackTrace();

//用户名不存在，跳转到登录页面

**this**.addActionError("用户名不存在！");

**return** "login";

}**catch** (IncorrectCredentialsException e) {

// 密码错误，跳转到登录页面

**this**.addActionError("密码错误！");

e.printStackTrace();

**return** "login";

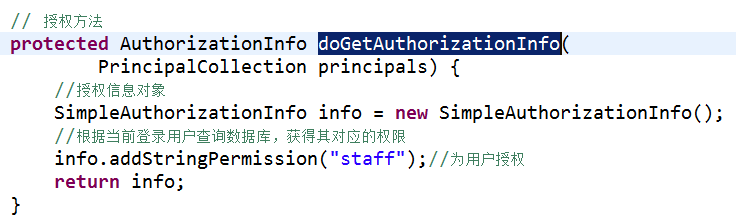
}

**return** "home";

}

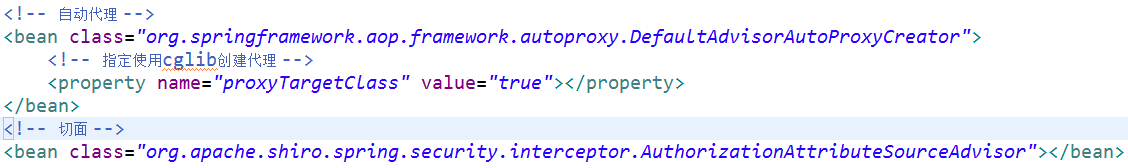
### 第五步：自定义一个Realm，进行认证和授权操作



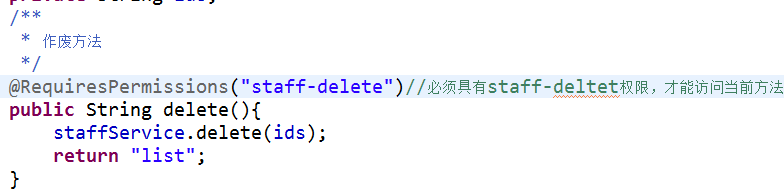


# Shiro提供的注解方式权限控制

### 第一步：在spring配置文件中配置自动代理和切面



### 第二步：在Action方法上使用注解



### 第三步：修改BaseAction的构造方法

