

**BOS物流项目**

# 教学计划

1、在realm中进行授权

2、使用shiro的方法注解方式权限控制

* + 在spring文件中配置开启shiro注解支持
  + 在Action方法上使用注解

3、使用shiro的标签进行权限控制

* + 在页面引入shiro的标签库
  + 在页面中使用标签

4、总结shiro提供的权限控制方式

* + URL拦截权限控制
  + 方法注解权限控制
  + 页面标签权限控制
  + 代码级别权限控制(了解)

5、权限管理

* + 初始化权限数据
  + 添加权限功能
  + 权限分页查询

# 在realm中进行授权

在BOSRealm中实现授权方法：



# 使用shiro的方法注解方式权限控制

第一步：在spring配置文件中开启shiro注解支持

<!-- 开启shiro框架注解支持 -->

<bean id=*"defaultAdvisorAutoProxyCreator"*

class=*"org.springframework.aop.framework.autoproxy.DefaultAdvisorAutoProxyCreator"*>

<!-- 必须使用cglib方式为Action对象创建代理对象 -->

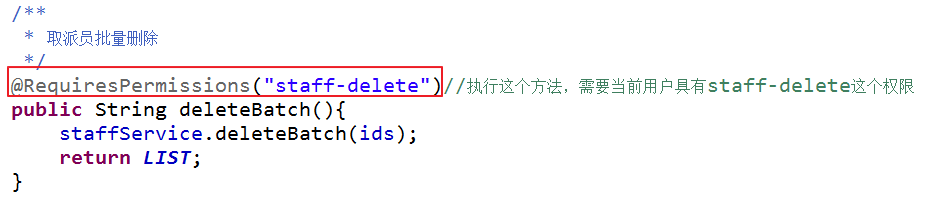
<property name=*"proxyTargetClass"* value=*"true"*/>

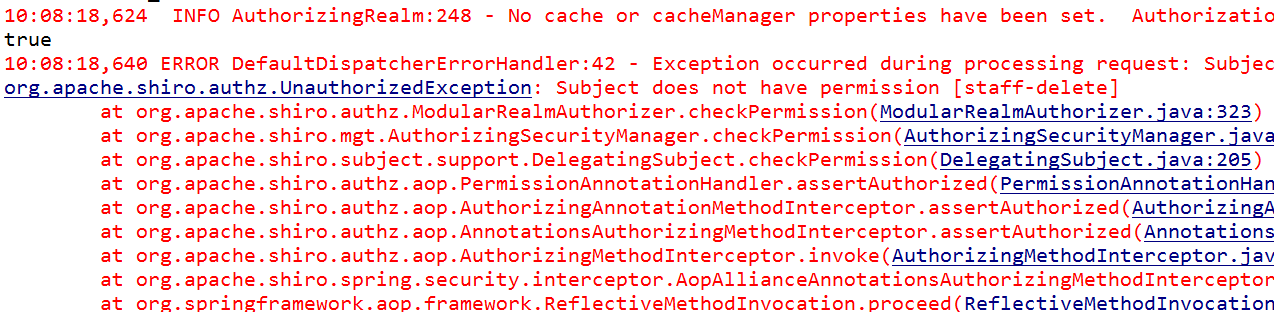
</bean>

<!-- 配置shiro框架提供的切面类，用于创建代理对象 -->

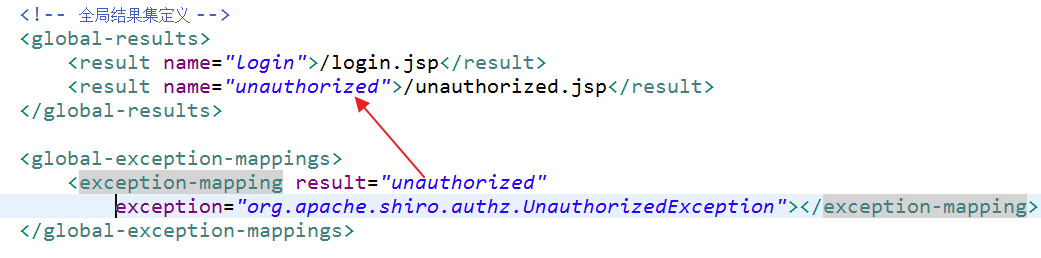
<bean class=*"org.apache.shiro.spring.security.interceptor.AuthorizationAttributeSourceAdvisor"*/>

第二步：在Action的方法上使用shiro注解





第三步：在struts.xml中配置全局异常捕获，当shiro框架抛出权限不足异常时，跳转到权限不足提示页面

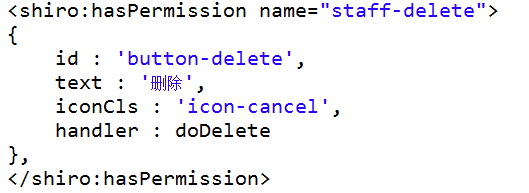


# 使用shiro提供的页面标签方式权限控制

第一步：在jsp页面中引入shiro的标签库

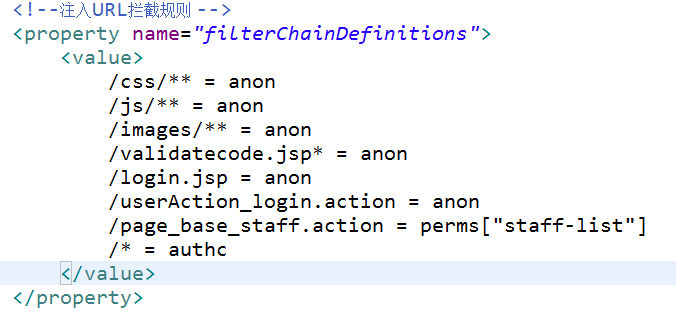
<%@ taglib prefix="shiro" uri="http://shiro.apache.org/tags" %>

第二步：使用shiro的标签控制页面元素展示

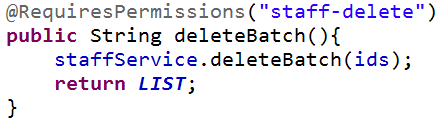


# 总结shiro框架提供的权限控制方式

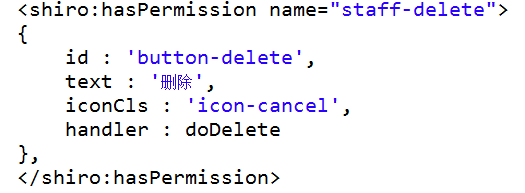
* URL拦截权限控制（基于过滤器实现）



* 方法注解权限控制（基于代理技术实现）



* 页面标签权限控制（标签技术实现）



* 代码级别权限控制（基于代理技术实现）

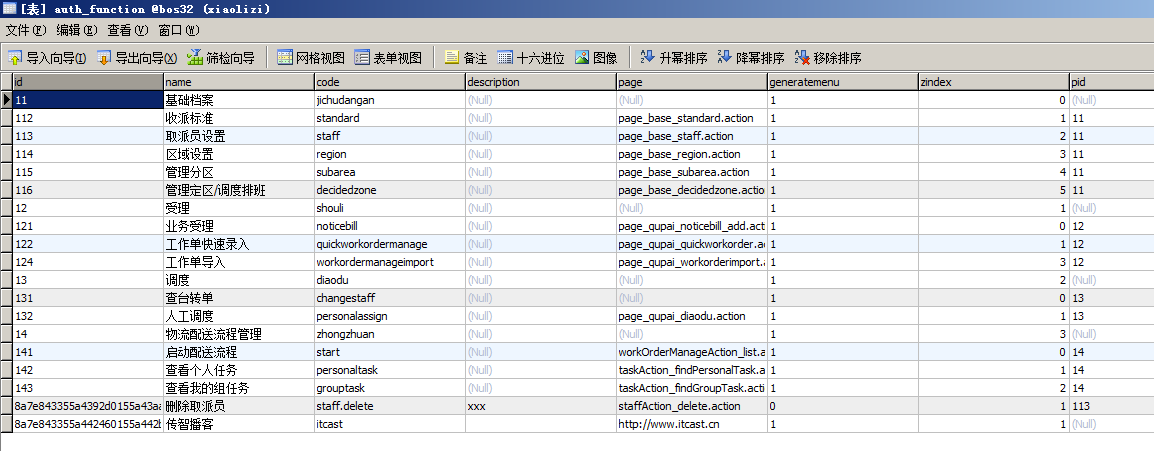


# 权限数据管理

## 初始化权限数据

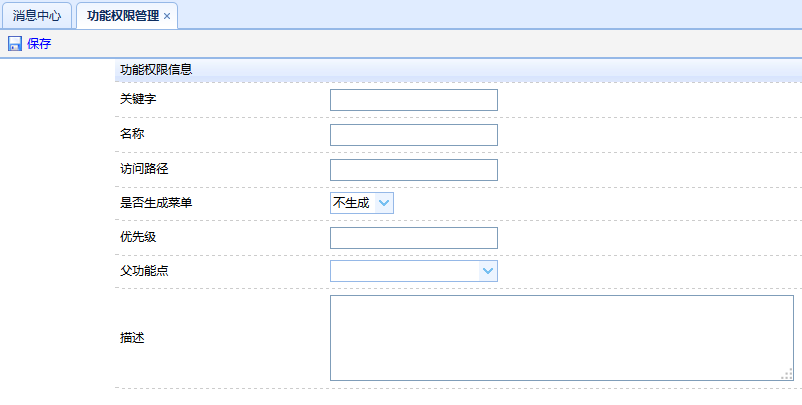
当项目上线后，如果正常运行，需要依赖一些基础数据支持。权限数据，就属于基础数据，因为系统的菜单是从权限表查询获得的。一般会提供sql脚本文件，导入基础数据。



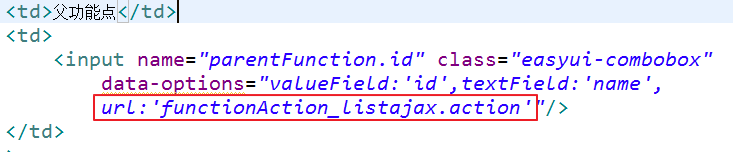


## 添加权限数据

页面：WEB-INF/pages/admin/function\_add.jsp

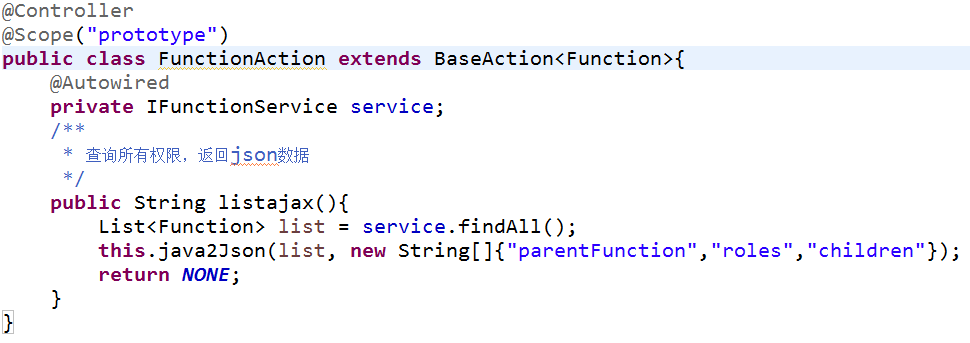


第一步：修改页面中父功能点对应的combobox，修改URL地址

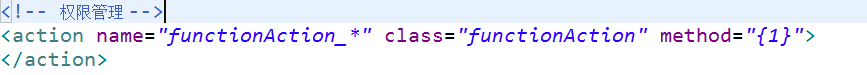


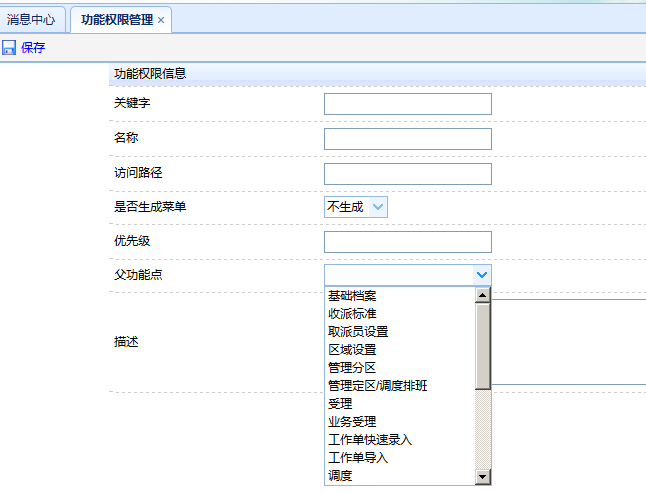


第二步：创建FunctionAction、Service、Dao，查询所有的权限，返回json

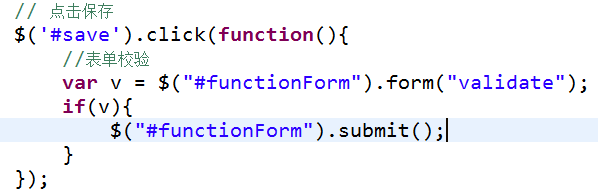


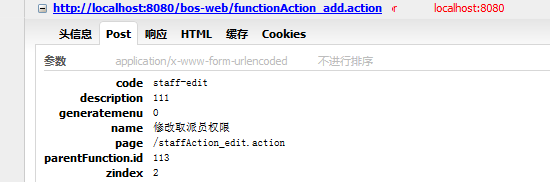
第三步：配置struts.xml



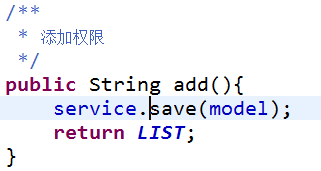


第四步：为添加页面中保存按钮绑定事件，进行校验，如果校验通过就提交表单

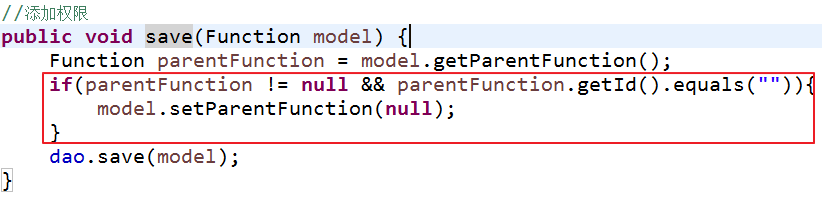




第五步：在FunctionAction中创建add方法，添加权限

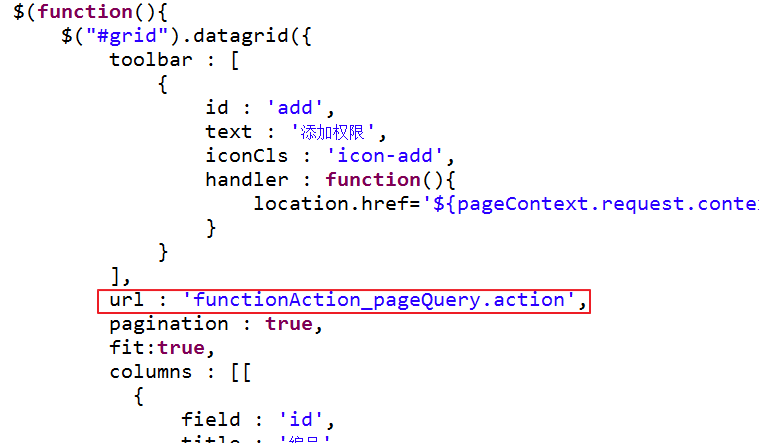


Service代码：



## 权限分页查询

第一步：修改页面中datagrid的URL地址



第二步：在Action中提供分页查询方法

