URL拦截设计

1、url拦截的控制器层为sysUrlPermissionControler，业务层sysUrlPermissionService，dao层为sysUrlPermission.map.xml，视图层为sysUrlPermission\*.jsp

2、url拦截数据库设计

sys\_url\_permission (url地址拦截管理)

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段类型 | 备注 |
| ID\_ | NUMBER(18) | 主键 |
| DESCP\_ | VARCHAR2(200) | 描述 |
| URL\_ | VARCHAR2(2000) | 拦截地址 |
| PARAMS\_ | VARCHAR2(500) | 拦截参数 |
| ENABLE\_ | NUMBER(0,0) | 是否启用0:禁用,1:启用 |

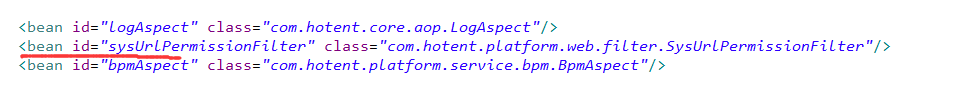
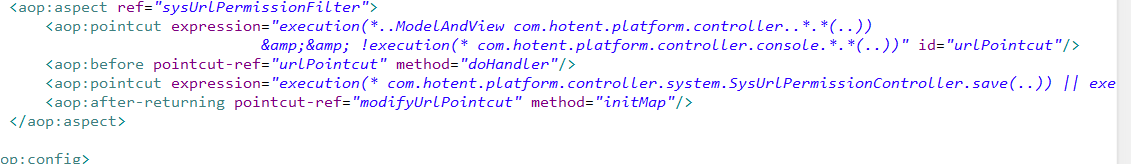
sys\_url\_rules (url地址拦截脚本管理)

|  |  |  |
| --- | --- | --- |
| 字段名 | 字段类型 | 备注 |
| ID\_ | NUMBER(18) | 主键 |
| SCRIPT\_ | CLOB | groovy脚本 |
| ENABLE\_ | NUMBER(0,0) | 是否启用 |
| SYS\_URL\_ID\_ | NUMBER(18) | URL地址拦截管理ID |
| DESCP\_ | VARCHAR2(500) | 描述 |
| SORT\_ | NUMBER(0,0) | 排序 |

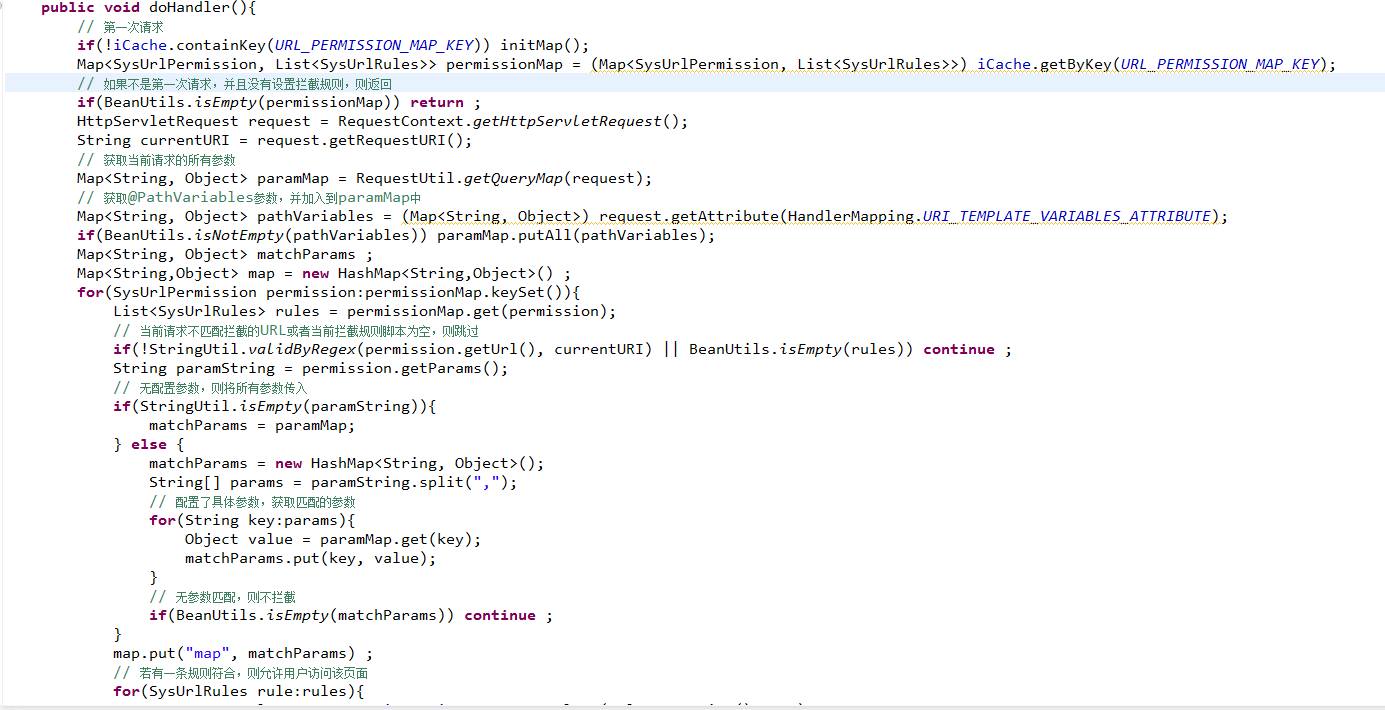
在系统中，我们把拦截脚本看作是对应拦截器的子表。

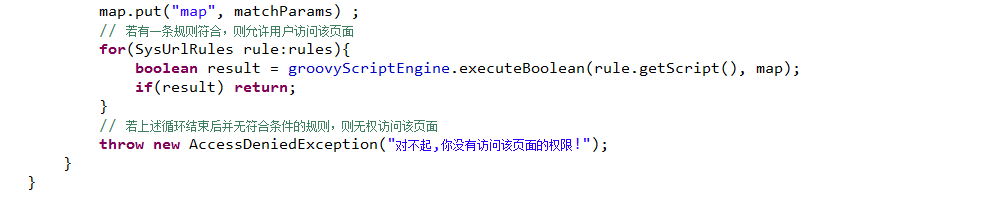
3、url拦截配置文件

在app-resource.xml文件里面配置了一个url拦截的过滤器sysUrlPermissionFilter。

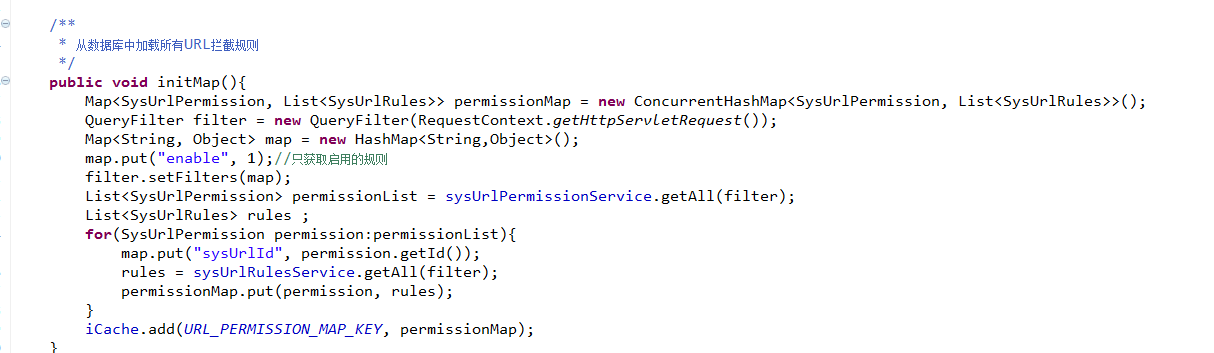
 

1. doHandler()方法





1. initMap()方法



4、url拦截的配置页面，在这个页面可以添加多个拦截器



点击编辑页面进入拦截器的编辑页面，在这个页面里面，url地址表示我们要拦截的url请求地址，参数名称表示要拦截上述【URL地址】中的哪些参数，如果要拦截所有的参数，则不需要填写。



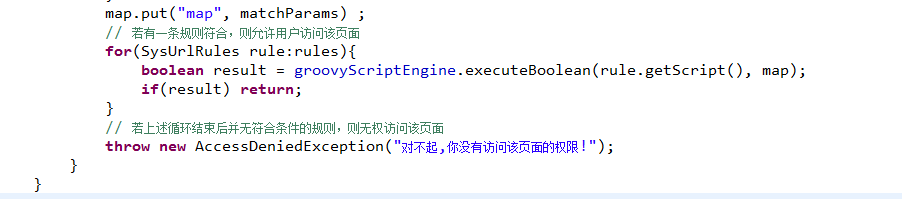
3、拦截规则

1. 可以编写多条规则，按优先级排序，优先级越高越先匹配
2. 规则脚本中可以使用map变量获取【参数名称】所配置的url参数，比如map.get("userId")<br>规则脚本需要返回boolean值，如下：

import com.hotent.core.util.ContextUtil;

return ContextUtil.getCurrentUserId().toString().equals(map.get("userId"));

1. 如果有多个规则，则会检查所有规则的返回值是否为true，如果有一个返回值为true，则表示可以访问这个url请求。



1. 规则的编辑页面

