**京东内网**

**单点登录及权限控制接入指南**

**V1.0**

## 简介

内网用户在单点登录ssa.jd.com登录后，单点登录服务会在.jd.com域中写入cookie:sso.jd.com来标识用户已登录，应用通过拦截器调用单点登录服务的JSF接口校验cookie:sso.jd.com值的有效性，正确无误时返回当前登录用户的信息，失败时返回自定义异常SsoException。

权限控制是业务系统的限控制资源码在统一身份管理系统（UIM）中注册，由UIM管理资源码、账号、角色、岗位。业务系统在功能中调用UIM的接口判断当前用户是否有访问权限，避免应用系统开发资源管理、角色管理、权限管理等功能，简化权限控制。

接入技术支持：叶昌伦、温树庭

## 环境信息

单点登录线上环境登录地址：<http://ssa.jd.com/sso/login>

单点登录测试环境登录地址：<http://test.ssa.jd.com/sso/login> 测试环境的密码为:xinxibu456

权限控制线环境上控制台地址：<http://idm.jd.com/jim>

权限控制测环境试控制台地址：<http://test.uim.jd.com/jim>

## 单点登录

### maven引入

|  |
| --- |
| <**dependency**>  <**groupId**>com.jd.common</**groupId**>  <**artifactId**>sso-uim-spring</**artifactId**>  <**version**>1.0.0-SNAPSHOT</**version**>  </**dependency**> |

如果和你的应用中的jar有冲突，请使用**exclusions**排除不需要的jar。

### Properties配置项

请将文本框中的配置信息通过**org.springframework.beans.factory.config.PropertyPlaceholderConfigurer**加载:

|  |
| --- |
| *#单点登录cookie,需要webapp.domain.name的域名保持对应* **sso.auth.cookie.domain**=**.jd.com**  *#可选配置，不需要登录可以访问的路径,以这个路径开头的地址不拦截，多个路径以逗号分割* **sso.exclude.path**=/**static,/rest** *#应用的域名* **webapp.domain.name**=**demo.jd.com** *#单点登录的登录地址,线上环境为http://ssa.jd.com/sso/login* **login.address**=**http://test.ssa.jd.com/sso/login** |

### 使用Spring MVC 3框架应用配置

加载单点登录的JSF服务接口，在spring配置文件中增加：

<**import resource="classpath\*:spring-sso-springmvc.xml"**/>

也可以自己直接在spring 配置文件中配置下文（注意init-method="initialize"不能缺少）：

|  |
| --- |
| *<!-- 单点登录JSF服务接口 -->*  <**jsf:consumer id="ssoService"  interface="com.jd.ssa.service.SsoService" protocol="jsf"  alias="v1" timeout="10000"**/>  *<!-- 单点登录拦截器配置 -->* <**mvc:interceptors**>  <**mvc:interceptor**>  <**mvc:mapping path="/\*\*"**/>  *<!-- 单点登录springmvc 拦截器bean -->* <**bean id="springSSOInterceptor" class="com.jd.common.springmvc.interceptor.SpringSSOInterceptor"  init-method="initialize"**>  <**property name="appDomainName" value="${webapp.domain.name}"**/>  <**property name="appHomeUrl" value="http://${webapp.domain.name}/"**/>  <**property name="loginUrl" value="${login.address}"**/>  <**property name="ssoDomainName" value="${sso.auth.cookie.domain}"**/>  <**property name="excludePath" value="${sso.exclude.path}"**/>  <**property name="ssoService" ref="ssoService"**/> </**bean**>  </**mvc:interceptor**>  </**mvc:interceptors**> |

### 使用Spring MVC 2框架应用配置

Spring 2 可使用多种URL匹配Bean方式：

|  |
| --- |
| org.springframework.web.servlet.handler.BeanNameUrlHandlerMapping |
| org.springframework.web.servlet.handler.SimpleUrlHandlerMapping |
| org.springframework.web.servlet.mvc.support.ControllerClassNameHandlerMapping |
| org.springframework.web.servlet.mvc.annotation.DefaultAnnotationHandlerMapping |

在配置这些Bean时将拦截器注入到**interceptors**属性中即可，下面是使用注解匹配URL方式的配置：

|  |
| --- |
| *<!-- 单点登录JSF服务接口 -->* <**jsf:consumer id="ssoService"  interface="com.jd.ssa.service.SsoService" protocol="jsf"  alias="v1" timeout="10000"**/>  <**bean id="springSSOInterceptor" class="com.jd.common.springmvc.interceptor.SpringSSOInterceptor"  init-method="initialize"**>  <**property name="appDomainName" value="${webapp.domain.name}"**/>  <**property name="appHomeUrl" value="http://${webapp.domain.name}/"**/>  <**property name="loginUrl" value="${login.address}"**/>  <**property name="ssoDomainName" value="${sso.auth.cookie.domain}"**/>  <**property name="excludePath" value="${sso.exclude.path}"**/>  <**property name="ssoService" ref="ssoService"**/> </**bean**>  <**bean class="org.springframework.web.servlet.mvc.annotation.DefaultAnnotationHandlerMapping"**>  <**property name="interceptors"**>  <**list**>  <**ref bean="springSSOInterceptor"** />  </**list**>  </**property**> </**bean**> |

### 使用Struts2框架应用配置

#### Bean配置

加载单点登录的JSF服务接口，在spring配置文件中增加：

<**import resource="classpath\*:spring-sso-struts.xml"**/>

也可以自己直接在spring 配置文件中配置下文（注意init-method="initialize"不能缺少）：

|  |
| --- |
| *<!-- 单点登录JSF服务接口 -->*  <**jsf:consumer id="ssoService"  interface="com.jd.ssa.service.SsoService" protocol="jsf"  alias="v1" timeout="10000"**/> *<!-- 单点登录struts2 拦截器bean -->* <**bean id="strutsSSOInterceptor" class="com.jd.common.struts.interceptor.StrutsSSOInterceptor"  init-method="initialize"**>  <**property name="ssoDomainName" value="${sso.auth.cookie.domain}"**/>  <**property name="appDomainName" value="${webapp.domain.name}"**/>  <**property name="appHomeUrl" value="http://${webapp.domain.name}/"**/>  <**property name="loginUrl" value="${login.address}"**/>  <**property name="excludePath" value="${sso.exclude.path}"**/>  <**property name="ssoService" ref="ssoService"**/> </**bean**> |

#### Struts2拦截器配置

在struts-config.xml中**package下的interceptors**配置下增加单点登录拦截器（注意此处的class是在spring中定义的bean id）：

|  |
| --- |
| <**interceptors**>  <**interceptor name="ssoInterceptor" class="strutsSSOInterceptor"**/>  *<!-- 单点登录 -->* <**interceptor-stack name="erpInterceptor"**>  <**interceptor-ref name="strutsDefaultStack"**/>  <**interceptor-ref name="ssoInterceptor"**/> </**interceptor-stack**>  </**interceptors**> |

增加全局result配置：

|  |
| --- |
| <**global-results**>  *<!-- 单点登录地址 -->* <**result name="login" type="redirect"**>${loginUrl}</**result**>  </**global-results**> |

在struts的package配置中使用：<**default-interceptor-ref name="erpInterceptor"**/>

或extends方式使用单点登录拦截器。

### 非Java应用配置

非Java语言开发的应用需要自己实现拦截器或直接调用接口校验用户、密码实现用户登录，可以使用的http接口有:

校验用户密码：

|  |  |
| --- | --- |
| **URL** | http://test.ssa.jd.com/sso/verify |
| **入参说明** | username 用户ERP账号  password 用户密码、需要使用md5加密 |
| **返回值** | 验证通过返回用户信息，REQ\_FLAG:true表示用户、密码匹配，失败请看REQ\_CODE和REQ\_MSG信息  {"REQ\_CODE":1,"REQ\_DATA":{"username":"yechanglun","email":"yechanglun@jd.com","mobile":"18910082090","orgId":"","personId":"","fullname":"叶昌伦","hrmDeptId":"12","hrmOrgId":"709","orgName":"","userId":226602},"REQ\_FLAG":true,"REQ\_MSG":"用户密码校验成功"} |

校验登录凭证ticket是否有效：

|  |  |
| --- | --- |
| **URL** | http://test.ssa.jd.com/sso/ticket/verifyTicket |
| **入参说明** | ticket 从cookie中读取的sso.jd.com值  url 当前用户访问应用URL地址  ip 当前用户的IP |
| **返回值** | 验证通过返回用户信息，REQ\_FLAG:true表示ticket有校，失败请看REQ\_CODE和REQ\_MSG信息  {"REQ\_CODE":1,"REQ\_DATA":{"username":"yechanglun","email":"yechanglun@jd.com","mobile":"18910082090","orgId":"","personId":"","fullname":"叶昌伦","hrmDeptId":"12","hrmOrgId":"709","orgName":"","userId":226602},"REQ\_FLAG":true,"REQ\_MSG":””}  失败时的返回值：  {"REQ\_CODE":3,"REQ\_DATA":{},"REQ\_FLAG":false,"REQ\_MSG":"无效ticket,请重新登录!"} |

获取正式登录凭证ticket:

当应用是非.jd.com的域名时，单点登录服务器无法将登录标识写入应用的cookie中，单点登录服务器会在用户登录成功返回应用的URL时带上临时标识：sso\_service\_ticket，应用调用接口获取正式的登录凭证写入到应用的cookie中，cookie名为:sso.jd.com,获取到ticket后就可以调用<http://test.ssa.jd.com/sso/ticket/verifyTicket>校验ticket有效性并返回用户信息。

|  |  |
| --- | --- |
| **URL** | http://test.ssa.jd.com/sso/ticket/**getTicket** |
| **入参说明** | sso\_service\_ticket 从http请求中中读取的参数，只能验证一次，30秒有效  url 当前用户访问应用URL地址  ip 当前用户的IP |
| **返回值** | 验证通过返回用户信息，REQ\_FLAG:true获取ticket成功，失败请看REQ\_CODE和REQ\_MSG信息  {"REQ\_CODE":1,"REQ\_DATA":{"25490f5fc60c4eff9bc73a4f13d52ce0”},"REQ\_FLAG":true,"REQ\_MSG":””} |

### 排除路径

对于不需要用户登录即可访问的静态问文件和对外提供服务的功能，可以通过在单点登录拦截器Bean中注入**excludePath**属性，来排除拦截，如有多个需要排除的地址，以英文都号分割这些地址;排除路径使用startsWith方式判断排除路径，配置的排除路径要以/开头，如：/static,/rest,/ws/service。

对于静态文件也可以通过在web.xml中配置servlet-mapping，让springmvc或struts2不拦截，如：

|  |
| --- |
| *<!-- ======== 排除静态文件 ======== --> <!-- 静态文件访问URL不会进入springmvc -->* <**servlet-mapping**>  <**servlet-name**>default</**servlet-name**>  <**url-pattern**>\*.jpg</**url-pattern**> </**servlet-mapping**> <**servlet-mapping**>  <**servlet-name**>default</**servlet-name**>  <**url-pattern**>\*.png</**url-pattern**> </**servlet-mapping**> <**servlet-mapping**>  <**servlet-name**>default</**servlet-name**>  <**url-pattern**>\*.jpeg</**url-pattern**> </**servlet-mapping**> <**servlet-mapping**>  <**servlet-name**>default</**servlet-name**>  <**url-pattern**>\*.js</**url-pattern**> </**servlet-mapping**> <**servlet-mapping**>  <**servlet-name**>default</**servlet-name**>  <**url-pattern**>\*.css</**url-pattern**> </**servlet-mapping**> <**servlet-mapping**>  <**servlet-name**>default</**servlet-name**>  <**url-pattern**>\*.ico</**url-pattern**> </**servlet-mapping**> |

### 单点退出

应用将页面转跳到单点登录服务的/sso/logout页面,单点系统将注销登录凭证（非.jd.com域名的应用请先调用com.jd.ssa.utils.SSOHelper类的logout方法后再转跳到单点退出页面/sso/logout?ReturnUrl=应用的URL）:

测试环境地址为：test.ssa.jd.com/sso/logout?ReturnUrl=应用的URL

线上环境地址为：ssa.jd.com/sso/logout?ReturnUrl=应用的URL

### 注意事项

1、当前登录用户信息获取方式：com.jd.common.web.LoginContext，这个类在很多Jar中都有，不要引用错误。

2、在应用各个模块的pom.xml中移除对erp-sso的依赖，这个包和sso-uim-spring有冲突。

3、<jsf:registry id="jsfRegistry" protocol="jsfRegistry" index="i.jsf.jd.com"/>应用中只能启动一次jsf注册中心，如果你的应用已使用JSF请确保是SSO加载前启动。

## 权限控制

应用系统的权限判断依赖在UIM系统中注册的资源码，权限管理员把资源码分配给角色、岗位。当用户在某一岗位时或拥有某些角色时就拥有这个岗位和这些角色的资源码。

如果应用的资源码是通过ERP的WebService（http://erp1.360buy.com/hrmservice/DeptWebService.asmx）中的ResourceNames方法获取的表示你的应用权限已在UIM做集中控制，在配置**com.jd.common.hrm.HrmPrivilegeHelper**时不用配置uimKey和uimToken二个属性，默认会读取ERP下的权限数据。否则需要先在UIM中注册你的应用、资源码、并授权后才能调用权限控制功能。

### Properties配置项

使用spring加载下文的配置信息

|  |
| --- |
| *#UIM中注册的app\_key* **uim.key**=**260b8d665e81494e85ff6d79af712291** *#UIM中注册的token* **uim.token**=**cbd5559d1a7e4825b123ca5258040a9d** *#uim的接口地址，线上环境为:http://uim.jd.local/opaauth/router/rest* **uim.url** = **http://test.uim.jd.com/opaauth/router/rest** |

### 编码拦截

在spring中加载以下配置即可使用：

|  |
| --- |
| *<!—权限判断Bean -->*  <**bean id="hrmPrivilegeHelper" class="com.jd.common.hrm.HrmPrivilegeHelper"**>  <**property name="uimKey" value="${uim.key}"**/>  <**property name="uimToken" value="${uim.token}"**/>  <**property name="uimUrl" value="${uim.url}"**/> </**bean**> |

在代码中调用hrmPrivilegeHelper.hasHrmPrivilege(code)方式判断用户是否有资源权限。

### 注解方式

通过在方法前增加注解方式来判断用户是否有权限操作，没有权限时返回http error code 401。注：spring web 2.5不支持注解方法判断权限。

|  |
| --- |
| @RequestMapping(**"add.html"**) @Authorization(**"PORTAL\_INFO\_ADMIN"**) **public** ModelAndView add() {  ModelAndView view = **new** ModelAndView(**"add"**);  **return** view; } |

应用可以在web.xml中配置error-page来显示更友好的无权限提示。

|  |
| --- |
| <**error-page**>  <**error-code**>401</**error-code**>  <**location**>/WEB-INF/401.jsp</**location**> </**error-page**> |

在error-page中可以通过<%=(String) request.getAttribute("javax.servlet.error.message")%>

方式获取错误信息。

当通过.json方式访问时无权限会返回json格式的无权限数据:

{"body":"accessDenied","status":false,"msg":"Access denied,code: **PORTAL\_INFO\_ADMIN**"}

#### Sping MVC3框架应用配置

注意拦截器配置顺序，权限拦截器要放在单点登录拦截器下面。

|  |
| --- |
| *<!—权限判断Bean -->*  <**bean id="hrmPrivilegeHelper" class="com.jd.common.hrm.HrmPrivilegeHelper"**>  <**property name="uimKey" value="${uim.key}"**/>  <**property name="uimToken" value="${uim.token}"**/>  <**property name="uimUrl" value="${uim.url}"**/> </**bean**> <**mvc:interceptors**>  *<!—单点登录拦截器 -->*  <**mvc:interceptor**>  <**mvc:mapping path="/\*\*"**/>  *<!-- 单点登录springmvc 拦截器bean -->*  <**bean id="springSSOInterceptor" class="com.jd.common.springmvc.interceptor.SpringSSOInterceptor"  init-method="initialize"**>  <**property name="appDomainName" value="${webapp.domain.name}"**/>  <**property name="appHomeUrl" value="http://${webapp.domain.name}/"**/>  <**property name="loginUrl" value="${login.address}"**/>  <**property name="ssoDomainName" value="${sso.auth.cookie.domain}"**/>  <**property name="excludePath" value="${sso.exclude.path}"**/>  <**property name="ssoService" ref="ssoService"**/> </**bean**>  </**mvc:interceptor**>  *<!—权限拦截器 -->*  <**mvc:interceptor**>  <**mvc:mapping path="/\*\*"**/>  *<!—权限拦截springmvc 拦截器bean -->* <**bean id="springAuthorizationInterceptor"  class="com.jd.common.springmvc.interceptor.SpringAuthorizationInterceptor"**>  <**property name="hrmPrivilegeHelper" ref="hrmPrivilegeHelper"**/> </**bean**>  </**mvc:interceptor**> </**mvc:interceptors**> |

#### Struts2框架应用配置

Spring Bean配置：

|  |
| --- |
| *<!—权限判断Bean -->*  <**bean id="hrmPrivilegeHelper" class="com.jd.common.hrm.HrmPrivilegeHelper"**>  <**property name="uimKey" value="${uim.key}"**/>  <**property name="uimToken" value="${uim.token}"**/>  <**property name="uimUrl" value="${uim.url}"**/> </**bean**>  *<!—Struts2 权限拦截器Bean -->* <**bean id="strutsAuthorizationInterceptor" class="com.jd.common.struts.interceptor.StrutsAuthorizationInterceptor"**>  <**property name="hrmPrivilegeHelper" ref="hrmPrivilegeHelper"**/> </**bean**> |

Struts-config.xml配置（权限拦截器要放在单点登录拦截器下面）:

|  |
| --- |
| <**interceptors**>  <**interceptor name="authorizationInterceptor" class="strutsAuthorizationInterceptor"**/>  <**interceptor-stack name="erpInterceptor"**>  <**interceptor-ref name="strutsDefaultStack"**/>  <**interceptor-ref name="ssoInterceptor"**/>  <**interceptor-ref name="authorizationInterceptor"**/> </**interceptor-stack**>  </**interceptors**> |

### 资源码缓存

资源码缓存为可选配置，没有配置缓存时会每次调用UIM的接口读取数据，如要提高系统响应速度可以在应用中增加资源码缓存配置，在Spring Bean中配置：

|  |
| --- |
| *<!—Jimdb配置，如应用自己已加载可以不用配置 -->*  <**bean id="jimClient" class="com.jd.jim.cli.ReloadableJimClientFactoryBean"**>  <**property name="jimUrl" value="${redis.url}"**/> </**bean**> <**bean id="redisClient" class="com.jd.common.cache.RedisClient"**>  <**property name="jimClient" ref="jimClient"**/> </**bean**> *<!—权限判断Bean -->*  <**bean id="hrmPrivilegeHelper" class="com.jd.common.hrm.HrmPrivilegeHelper"**>  <**property name="uimKey" value="${uim.key}"**/>  <**property name="uimToken" value="${uim.token}"**/>  <**property name="uimUrl" value="${uim.url}"**/>  <**property name="redisClient" ref="redisClient"**/> </**bean**> |

将jimdb的com.jd.jim.cli.Cluster对象注入com.jd.common.cache.RedisClient 的jimClient属性;

默认缓存60\*15秒，需要修改时请将缓存时长（秒）注入**hrmPrivilegeHelper**的**cacheTime**属性

## 参考配置：

Authen.properties

|  |
| --- |
| *#应用的域，常见值为.jd.com或.360buy.com*  **sso.auth.cookie.domain**=**.jd.com** *#不需要登录可以访问的路径* **sso.exclude.path**=**/static,/rest**  *#应用的域名* **webapp.domain.name**=**demo.jd.com**  *#单点登录地址* **login.address**=**http://test.ssa.jd.com/sso/login** *#UIM中注册的app\_key，需要权限拦截时配置* **uim.key**=**260b8d665e81494e85ff6d79af712291** *#UIM中注册的token，需要权限拦截时配置* **uim.token**=**cbd5559d1a7e4825b123ca5258040a9d** *#uim的接口地址，线上环境为:http://uim.jd.local/opaauth/router/rest，需要权限拦截时配置* **uim.url** = **http://test.uim.jd.com/opaauth/router/rest** *#jimdb配置* **redis.url**=**jim://1803528818953446384/1** |

Spring MVC接入配置：

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:jsf="http://jsf.jd.com/schema/jsf"  xmlns:mvc="http://www.springframework.org/schema/mvc"  xsi:schemaLocation="  http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.2.xsd  http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc.xsd  http://jsf.jd.com/schema/jsf http://jsf.jd.com/schema/jsf/jsf.xsd"  default-autowire="byName"**>  *<!--  <jsf:registry id="jsfRegistry" protocol="jsfRegistry" index="i.jsf.jd.com"/>  -->*  *<!-- 单点登录JSF服务接口 -->* <**jsf:consumer id="ssoService"  interface="com.jd.ssa.service.SsoService" protocol="jsf"  alias="v1" timeout="10000"**/>  *<!—Jimdb配置，如应用自己已加载可以不用配置 -->*  <**bean id="jimClient" class="com.jd.jim.cli.ReloadableJimClientFactoryBean"**>  <**property name="jimUrl" value="${redis.url}"**/>  </**bean**>  *<!—缓存调用代理类 -->*  <**bean id="redisClient" class="com.jd.common.cache.RedisClient"**>  <**property name="jimClient" ref="jimClient"**/>  </**bean**>  *<!—权限判断Bean -->*  <**bean id="hrmPrivilegeHelper" class="com.jd.common.hrm.HrmPrivilegeHelper"**>  <**property name="uimKey" value="${uim.key}"**/>  <**property name="uimToken" value="${uim.token}"**/>  <**property name="uimUrl" value="${uim.url}"**/>  <**property name="redisClient" ref="redisClient"**/>  </**bean**>  <**mvc:interceptors**>  <**mvc:interceptor**>  <**mvc:mapping path="/\*\*"**/>  *<!— Spring mvc 单点登录拦截器bean -->*  <**bean id="springSSOInterceptor" class="com.jd.common.springmvc.interceptor.SpringSSOInterceptor"  init-method="initialize"**>  <**property name="appDomainName" value="${webapp.domain.name}"**/>  <**property name="appHomeUrl" value="http://${webapp.domain.name}/"**/>  <**property name="loginUrl" value="${login.address}"**/>  <**property name="ssoDomainName" value="${sso.auth.cookie.domain}"**/>  <**property name="excludePath" value="${sso.exclude.path}"**/>  <**property name="ssoService" ref="ssoService"**/>  </**bean**>  </**mvc:interceptor**>  <**mvc:interceptor**>  <**mvc:mapping path="/\*\*"**/>  *<!— Spring mvc 权限拦截器bean -->*  <**bean id="springAuthorizationInterceptor"  class="com.jd.common.springmvc.interceptor.SpringAuthorizationInterceptor"**>  <**property name="hrmPrivilegeHelper" ref="hrmPrivilegeHelper"**/>  </**bean**>  </**mvc:interceptor**>  </**mvc:interceptors**> </**beans**> |

Struts2 Spring Bean配置：

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:jsf="http://jsf.jd.com/schema/jsf"  xsi:schemaLocation="  http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans-3.2.xsd  http://jsf.jd.com/schema/jsf   http://jsf.jd.com/schema/jsf/jsf.xsd"  default-autowire="byName"**>  *<!--  <jsf:registry id="jsfRegistry" protocol="jsfRegistry" index="i.jsf.jd.com"/>  -->*  *<!-- 单点登录JSF服务接口 -->* <**jsf:consumer id="ssoService"  interface="com.jd.ssa.service.SsoService" protocol="jsf"  alias="v1" timeout="10000"**/>  *<!—Jimdb配置，如应用自己已加载可以不用配置 -->*  <**bean id="jimClient" class="com.jd.jim.cli.ReloadableJimClientFactoryBean"**>  <**property name="jimUrl" value="${redis.url}"**/>  </**bean**>  *<!—缓存调用代理类 -->*  <**bean id="redisClient" class="com.jd.common.cache.RedisClient"**>  <**property name="jimClient" ref="jimClient"**/>  </**bean**>  *<!—权限判断Bean -->*  <**bean id="hrmPrivilegeHelper" class="com.jd.common.hrm.HrmPrivilegeHelper"**>  <**property name="uimKey" value="${uim.key}"**/>  <**property name="uimToken" value="${uim.token}"**/>  <**property name="uimUrl" value="${uim.url}"**/>  <**property name="redisClient " ref="redisClient"**/>  </**bean**>  *<!—单点登录 Struts2拦截器Bean -->* <**bean id="strutsSSOInterceptor" class="com.jd.common.struts.interceptor.StrutsSSOInterceptor"  init-method="initialize"**>  <**property name="ssoDomainName" value="${sso.auth.cookie.domain}"**/>  <**property name="appDomainName" value="${webapp.domain.name}"**/>  <**property name="appHomeUrl" value="http://${webapp.domain.name}/"**/>  <**property name="loginUrl" value="${login.address}"**/>  <**property name="ssoService" ref="ssoService"**/>  </**bean**>  *<!—权限 Struts2拦截器Bean -->*  <**bean id="strutsAuthorizationInterceptor" class="com.jd.common.struts.interceptor.StrutsAuthorizationInterceptor"**>  <**property name="hrmPrivilegeHelper" ref="hrmPrivilegeHelper"**/>  </**bean**> </**beans**> |

Struts-config.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="GBK"** *?>* **<!DOCTYPE struts PUBLIC  "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"  "http://struts.apache.org/dtds/struts-2.0.dtd"*>*** <**struts**>  <**constant name="struts.action.extension" value="html"**/>  <**constant name="struts.i18n.encoding" value="UTF-8"**/>  <**constant name="struts.custom.i18n.resources" value="i18n/messages"**/>  *<!-- 放过的地址 -->* <**constant name="struts.action.excludePattern" value="/js/\*"**></**constant**>  *<!-- Struts2 OGNL白名单-->* <**constant name="ognl.security.enabled" value="true"**/>  *<!-- 拦截器调用栈配置 -->* <**package name="webapp-default" extends="struts-default"**>  <**result-types**>  <**result-type name="velocity"  class="com.jd.common.struts.velocity.VelocityLayoutResult" default="true"**/>  </**result-types**>  <**interceptors**>  *<!—单点登录 Struts2拦截器 -->*  <**interceptor name="ssoInterceptor" class="strutsSSOInterceptor"**/>  *<!—权限管理 Struts2拦截器 -->*  <**interceptor name="authorizationInterceptor" class="strutsAuthorizationInterceptor"**/>  *<!-- 单点登录及权限拦截堆栈 -->* <**interceptor-stack name="erpInterceptor"**>  <**interceptor-ref name="strutsDefaultStack"**/>  <**interceptor-ref name="ssoInterceptor"**/>  <**interceptor-ref name="authorizationInterceptor"**/>  </**interceptor-stack**>  </**interceptors**>  <**global-results**>  *<!-- 单点登录地址 -->* <**result name="login" type="redirect"**>${loginUrl}</**result**>  </**global-results**>  <**global-exception-mappings**>  <**exception-mapping exception="java.lang.Exception" result="exception"**/>  </**global-exception-mappings**>  </**package**>   *<!-- erp 单点登录 -->* <**package name="erp-default" extends="webapp-default"**>  <**default-interceptor-ref name="erpInterceptor"**/>  </**package**>  <**package name="user" namespace="/user" extends="erp-default"**>  <**action name="index" class="userAction" method="index"**>  <**result name="success"**>/WEB-INF/vm/index.vm</**result**>  </**action**>  <**action name="list" class="userAction" method="list"**>  <**result name="success"**>/WEB-INF/vm/list.vm</**result**>  </**action**>  <**action name="add" class="userAction" method="add"**>  <**result name="success"**>/WEB-INF/vm/add.vm</**result**>  </**action**>  </**package**> </**struts**> |

