一、js文件引用说明

认证中心测试地址**http://www.sso.com:8080**

1. 登录的位置引用js文件

<**script id="login" th:src="@{http://www.sso.com:8080/js/sso-login.js}" redirectUrl="http://www.c1.com:8081/"**></**script**>

1. 退出按钮位置引用js文件

<**script id="logOut" th:src="@{http://www.sso.com:8080/js/sso-logOut.js}" redirectUrl="http://www.c1.com:8081/login"**></**script**>

3、引用js文件参数说明

<**script id="login" th:src="@{http://www.sso.com:8080/js/sso-login.js}" redirectUrl="http://www.c1.com:8081/"**></**script**>

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | **值** | **说明** | **是否可修改** |
| **id** | **Login/logOut** | **对应登录/登出标识** | **不可修改** |
| **src** | **<http://www.sso.com:8080/js/sso-login.js>或**  **http://www.sso.com:8080/js/sso-logOut.js** | **认证中心对应登录/登出js文件位置** | **不可修改** |
| **redirectUrl** | **http://www.c1.com:8081/index** | **登录/登出成功后跳转的地址** | **可修改** |

1. 对接系统拦截器流程

1、判断是否有局部的会话

HttpSession session = req.getSession();  
Boolean isLogin = (Boolean)session.getAttribute(**"isLogin"**);  
**if** (**null**!=isLogin && isLogin){

放行

}

2、判断地址栏中是否有携带token参数。

String token = req.getParameter(**"token"**);  
**if** (!StringUtils.*isEmpty*(token)){  
 *//token地址不为空 说明地址栏中包含了token，拥有令牌  
 //判断token是否有认证中心生成的* Connection.Response resp = Jsoup.*connect*( **"http://www.sso.com:8080/verify"**)  
 .data(**"token"**, token)  
 .data(**"clientUrl"**,**"**http://www.c1.com:8081**/logOut"**)  
 .data(**"jsessionid"**,session.getId())  
 .method(Connection.Method.***GET***).execute();  
 String isVerify = resp.body();  
 **if** (**"true"**.equals(isVerify)){  
 *//说明token是有统一认证中心产生的，可以创建局部的会话* session.setAttribute(**"isLogin"**,**true**);  
 *//放行该次请求* reqLogOut(req);  
 chain.doFilter(request, response);  
 **return**;  
 }  
}

1. 没有局部会话，重定向到统一认证中心，检查是否其他系统已经登录过

[http://www.sso.com:8080](http://www.sso.com:8080/checkLogin?redirectUrl=http:/www.c1.com:8081/xxx)**[/checkLogin?redirectUrl=](http://www.sso.com:8080/checkLogin?redirectUrl=http:/www.c1.com:8081/xxx)**[http://www.c1.com:8081/xxx](http://www.sso.com:8080/checkLogin?redirectUrl=http:/www.c1.com:8081/xxx)

4、需写一个退出方法供认证中心调用:接口名必须是 /logOut

*销毁全局会话*

1. 相关接口

认证中心地址：**<http://www.sso.com:8080>**

*1、对token是否由认证中心生成验证接口*

*接口： /verify*

*Method: post*

*参数*

|  |  |  |  |
| --- | --- | --- | --- |
| *字段* | *类型* | *非空* | *说明* |
| *token* | *字符串* | *是* | *从地址栏中得到的token* |
| *clientUrl* | *字符串* | *是* | *对接系统的退出接口地址* |
| *jsessionid* | *字符串* | *是* | *sessionid* |

*返回结果：true||false*

*2、检查是否其他系统已经登录过*

*接口：/checkLogin*

*Methoed:GET*

*参数*

|  |  |  |  |
| --- | --- | --- | --- |
| *字段* | *类型* | *非空* | *说明* |
| *redirectUrl* | *字符串* | *是* | *成功后重定向地址* |

*返回结果：成功：重定向到首页，失败：返回到登录页*

*3、获取用户登录信息接口*

*接口：/userInfo*

*Methoed:get*

*参数：无*

*返回结果：*

*成功{"code":200,"data":[内容待定],"msg":"成功"}*

*失败{"code":0,"data":null,"msg":"Token Can not be empty"}*