单点登录

目录

简介：

单点登录（Single Sign On），简称为 SSO，是目前比较流行的企业业务整合的解决方案之一。SSO 的定义是在多个应用系统中，用户只需要登录一次就可以访问所有相互信任的应用系统。

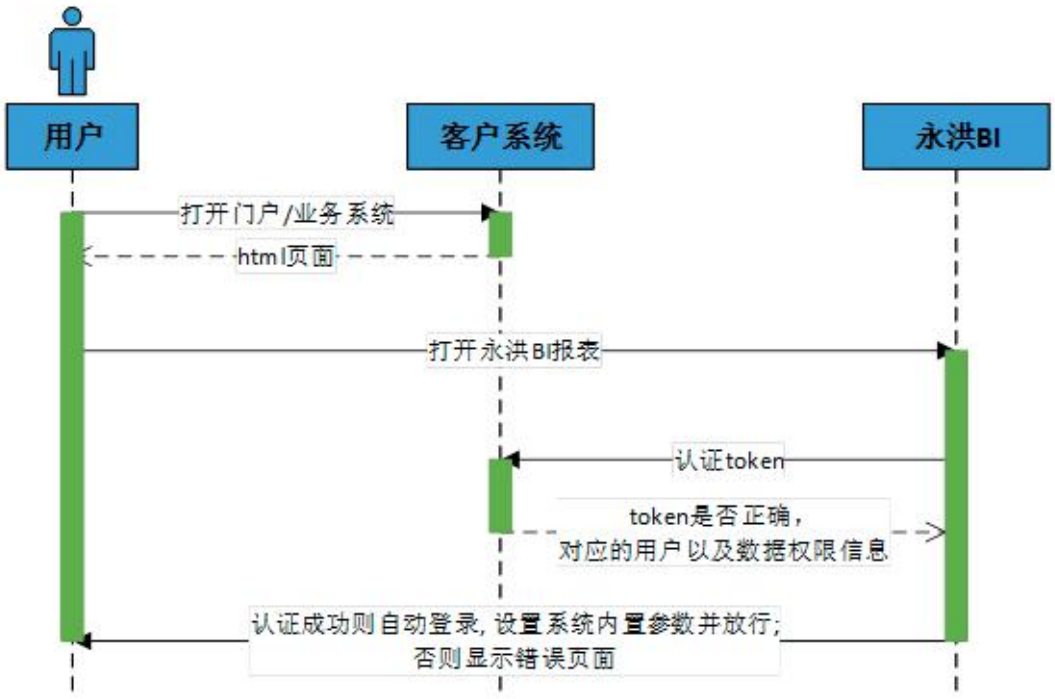
1、实现方式

1）永洪支持的标准单点登录是采用 token 验证的方式，下面详细介绍下这种方式。

2）token 验证原理说明

客户在自己的系统中登录以后，希望能直接打开永洪的报表或者功能模块，但是这个用户可能之前没有在永洪系统中创

建过。这种方式需要在报表或者功能模块的 url 后面附带 token 参数和 sysFlag 参数（sysFlag 非必要，该参数作用是为了区分多个系统 token 验证的回调接口，根据参数的不同选取不同的回调地址，如果没有该参数或者该参数为空字符串，会去读取默认配置的 url 回调接口），当点击报表或者模块的 url 时，永洪系统拦截这些请求，调用客户系统提供的回调接口（需要客户开发）验证 token 是否是合法生成的，验证通过后回调接口返回当前用户信息。永洪系统会判断当前用户是否已经存在，如果不存在，会创建该用户，创建好后就直接登录系统，以避免出现登录页面。



注：在第二步，返回的html页面中有包含永洪BI报表的iframe，并且永洪BI的链接中包含一个随机生成和用户相关连的字符串参数(token)

Token 和回调接口由客户系统生成，出于以下理由：

当用户从客户系统进入永洪系统中时可能在永洪系统中并不存在，这时候永洪需要从客户系统中获取当前用户信息，在永洪系统中自动创建，才能访问永洪系统。用 token 而不是直接传递用户名是为了安全考虑，需要先验证下。

Token 和回调接口需要满足以下条件：

根据当前登录用户的当前会话生成，每次都不一样，可以采用 hash 算法或者其他算法。

客户需要开发接口，来验证永洪回传的接口，以及根据 token 来返回当前登录用户相关信息。

3）Token 回调使用

请求地址	callback.url

请求参数(post 方式)	{
	token:E2ABA91383139F9D4B4D7C1E0226FA1B
	}

返回参数	{ "result": "success", "userId": "john", "userAlias": "john", "userEmail": "john@.com", "userRoles": "角色 1,角色 2", "userGroups": "组 1/组 1 子组,组 2/组 2 子组",

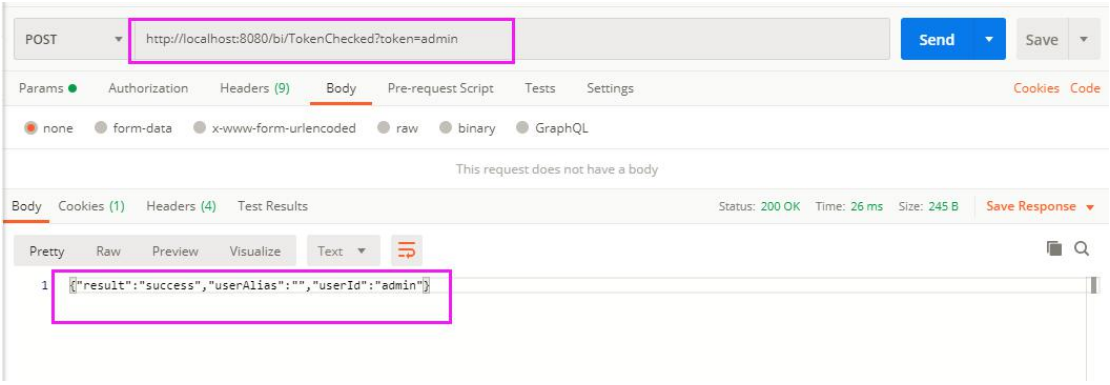
	"param":{
	"department": "总部",
	"city": "北京",
	"xxx": "xxx"
	}
	}

➢注:

回调验证 token 接口，可以使用 post 方式验证，参数名是 token。

在回调地址中拼上 token 进行验证，回调验证 token 接口验证成功，返回的结果中必须有 result 和 userId 。result 的值为 success 代表接口回调成功，如果返回其他值为失败。

如截图中 postman 工具的验证结果，其中 http\://localhost\:8080/bi/TokenChecked 为本地配置的回调。



其中，userId 即是永洪中的用户名。userAlias 可选，是用户别名。userEmail 可选，是用户邮箱。userRoles 可选，是用户角色，如果用户有多个角色以逗号分隔。userGroups 可选，

是用户组，如果用户属于多个组以逗号分隔，需写入组的全路径，多层级组以”/”分隔。param 可选，对应一个 json 对象，里面存储需要放到产品内置参数中的数据，可用来进行数据权限过滤。

4）token 传递方式

token 可以放在 url 后面，也可以通过 post 提交也可以放在 header 里面。放在 url 后面举例如：

http://localhost:8080/bi/Viewer?proc=1&token=E2ABA91383139F9D4B4D7C1E0226FA1B

5）客户的回调接口 java 代码举例

```
（不能直接使用，需要客户完善）
package com.customer.service;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.*;
public class TokenCheckServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;
	public TokenCheckServlet() {
		super();
	}
	protected void doGet(HttpServletRequest request,
		HttpServletResponse response) throws ServletException, IOException {
		String token = request.getParameter("token");
		if ((token == null) || "".equals(token.trim())) {
			//客户自己的错误处理逻辑
			return;
		}
		else {
			//  校验 token 是否合法
			StringBuilder responseStr = new StringBuilder();
			if (check(token)) {
				//token 验证通过就返回当前登录用户。对应永洪系统中的用户名。
				responseStr.append(
					"{\"result\":\"success\",\"userId\":\"test\"}");
			}
			else {
				//token 验证没通过就不返回 userId。
				responseStr.append("{\"error\":\"james\"}");
			}
```

2、标准单点登录部署样例

在完成集成后，进行单点登录配置，配置步骤如下：

1）在$安装目录/tomcat/webapps/bi/WEB-INF/web.xml

配置 Servlet（配置单点登录的拦截器。拦截产品请求，先进行单点登录的逻辑。）

需要添加的内容：

```
<!-- sso filter start -->
 <filter>
    <filter-name>StandardSSOFilter</filter-name>
    <filter-class>g5.sv.standardsso.SSOLoginFilter</filter-class>
 </filter>
 <filter-mapping>
    <filter-name>StandardSSOFilter</filter-name>
    <servlet-name>ViewerServlet</servlet-name>
 </filter-mapping>
```

2）配置 bi.properties 文件：

A、Yonghong/bihome/bi.properties 文件中添加以下属性 standardsso.callback.url=来验证 token 的回调地址链接。此链接由客户系统提供，用于验证 token 信息，比如：

standardsso.callback.url=http\://localhost\:8080/bi/TokenChecked

B、若客户系统有多个业务系统的回调 url 需配置 standardsso.callback.url.***=，****的值为传入 sysFlag 的值（详细描述见附录）。比如配置如下：

standardsso.callback.url.test1=http\://localhost\:8081/bi/TokenChecked

standardsso.callback.url.test2=http\://localhost\:8082/bi/TokenChecked

则系统访问时传入 sysFlag 值就为 test1 与 test2，如业务系统 1 单点登录访问 BI：http\://ip:port/bi/Viewer?sysFlag=test1&token=。

3）重启 tomcat

重启 tomcat，单点登录的配置即会生效。

3、附录

标准单点登录的配置项，如下表所示：

（如需使用，可在 bi.properties 中进行配置）

配置项	默认值	描述
standardsso.enabled	false	是否启用单点登录
standardsso.callback.url	无	客户系统提供的回调接口的 url，用于验证 token 是否合法
standardsso. callback.url. ****	无	客户系统提供的多个回调接口的 url，****的值为传入 sysFlag 的值。 根据不同的 sysFlag 选取相对应的回调 url。比如，传入的 sysFlag 为 test，那么会去读取 standardsso.callback.url.test 属性对应的值为回调 url，如果 sysFlag 为空，则会去默认读取 standardsso.callback.url 的地址。
standardsso. allowType	无	自动创建用户时需要给其设置访问永洪系统的权限，要不然什么都无权查看。有这些可选值：viewerDb,viewerManager,editor,query，conn，ml viewerDb：授予访问报表权限（适用于嵌入永洪报表到客户系统中时，只想给

		用户访问该报表的权限。系统会自动将 url 中的报表授予当前用户查看权限） viewerManager：授予查看报表功能模块权限（适用于嵌入永洪的功能模块到客户系统时，系统自动授予用户访问该功能模块的权限） editor：授予访问编辑报表模块权限（适用于嵌入永洪的功能模块到客户系统时，系统自动授予用户访问该功能模块的权限） query：授予访问新建查询模块权限（适用于嵌入永洪的功能模块到客户系统时，系统自动授予用户访问该功能模块的权限） conn：授权访问新建数据源模块权限（适用于嵌入永洪的功能模块到客户系统时，系统自动授予用户访问该功能模块的权限） ml :授权访问深度分析模块（适用于嵌入永洪的功能模块到客户系统时，系统自动授予用户访问该功能模块的权限） 可以给自动创建的用户配置 1 个或者

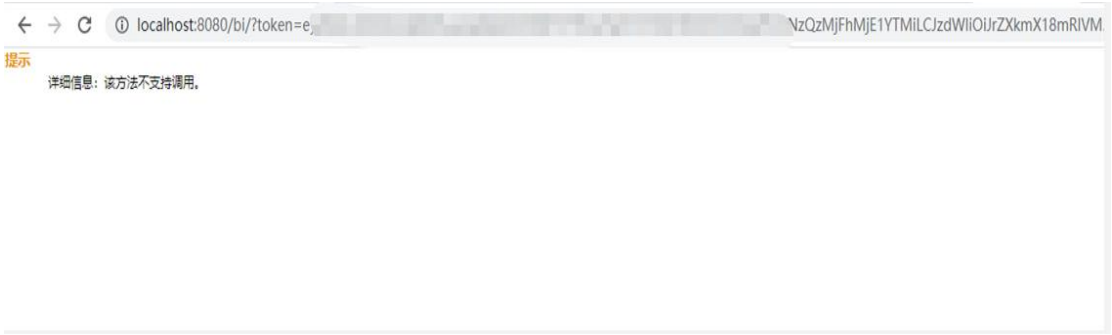
		多个权限，也可以不配置，当配置上之后，当永洪检测到用户没有权限访问这些模块时，会自动赋予查看权限。默认为空，即不授予任何权限。
standardsso. autoCreateUser	无	是否自动创建用户，只有 true 或者 false 两个可选值。选择 true 时，Token 接口返回来的用户在永洪中如果不存在，会自动创建到指定组中，新建用户的初始密码为 8 位随机数。选择 false 时，不会自动创建用户，会在页面提示用户不存在。
standardsso. saveUserDir	无	自动创建用户到某个用户组下。默认配置的组名是“单点登录“。（如果包含中文需要用 unicode 编码，纯英文不需要编码）注意：此用户组不需手动创建，系统会自动创建，建议都配置一个目录，这样方便以后查哪些用户是通过单点登录创建的。注意：不能有\\/:*?"<>| 特殊字符。
standardsso.autoUpdateGroup	false	是否自动更新用户所属组，只有 true 或者 false 两个可选值。选择 true 时，会根据接口返回来的用户组

		userGroups 参数，并且该参数不为空或空字符串时，则更新用户到组下面（组不存在会自动创建）。选择 false 时，不会根据该参数去更新用户所属的组信息。
standardsso.autoUpdateRole	false	是否自动更新用户的角色，只有 true 或者 false 两个可选值。选择 true 时，会根据接口返回来的用户角色 userRoles 参数，并且该参数不为空或空字符串，则更新用户的角色信息（角色不存在会自动创建）。选择 false 时，不会根据该参数去更新用户的角色信息。
standardsso.autoUpdateUser	false	是否自动更新用户信息，只有 true 或者 false 两个可选值。选择 true 时，会根据接口返回来的用户信息更新用户信息。选择 false 时，不会更新用户信息。
standardsso.anonymous.url	api, TokenChecked	不进行单点登录拦截的 url，当用户请求这些 url 时将直接放行，而不进行单点登录验证。 注意配置回调 url，例如：

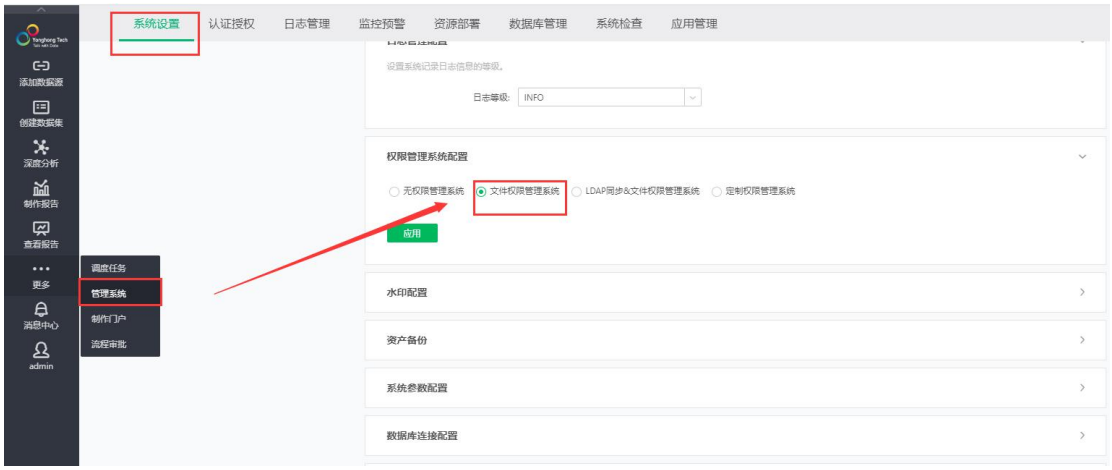
		api,TokenChecked
standardsso.token.invalid.jumpurl	无	如果 token 校验接口成功，但返回的状态不对，如果配置该跳转地址，则跳转到 url。

4、常见报错处理

1）、详细信息：该方法不支持调用



解决办法：请检查 BI 系统是否为文件权限管理系统。



2）、报错提示：回调 URL”standardsso.callback.url”不能为空。

解决方法：

（1）先检查永洪环境是文件系统还是数据库系。参考如下截图：

若是文件系统：回调地址的配置就在安装目录 Yonghong/bihome/bi/propertise 中添加，重启 BI 生效。

若是数据库系统：则 bi.propertise 文件需要在前台下载，添加回调地址的配置后再上传覆盖。

（2）Yonghong/bihome 下 bi.propertise 中有没有添加上述回调地址的配置。

3）token 校验失败。

原因：通常是回调接口有问题导致 BI 校验不通过，或者 BI 里回调地址添加有误。

解决办法：

（1）查看客户浏览器上个访问的 url 地址，若是拼接了 sysFlag 参数的多回调接口的校验，对应检查 Yonghong/bihome 下 bi.propertise 里回调地址是否正确，

（2）回调地址上拼上 token 查看返回的用户是否正常，未正常返回则客户回调接口有问题。