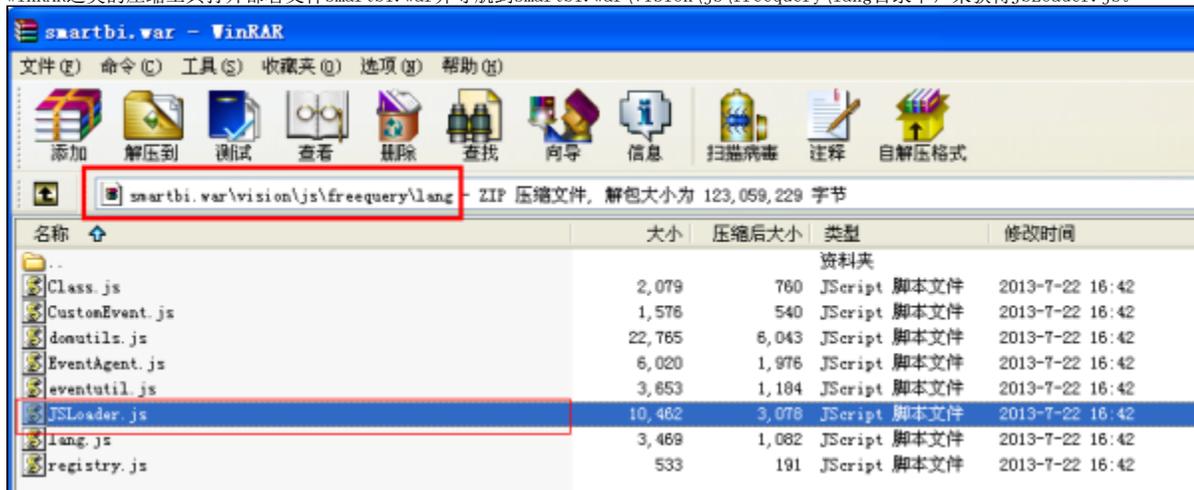


# 使用浏览器端SDK单点登录到Smartbi

在系统集成中，用户为了统一门户，一般都会要求多个系统实现单点登录。在第三方系统中通过编写 JavaScript 代码，调用 Smartbi 浏览器端SDK 中的方法进行登录即可。浏览器端SDK 通过 JavaScript API 提供多个客户端组件供外部调用，满足非 J2EE 平台的客户环境（如.Net平台）实现与 Smartbi 系统的集成要求。具体操作步骤如下：

1、首先部署Smartbi服务器。假设Smartbi已经部署到应用服务器中，访问地址为：<http://localhost:8080/smartbi>

2、将导航到smartbi\vision\js\freequery\lang\JSLoader.js拖放到本地文件夹中，然后将其添加到自己的项目中。如果使用的是smartbi.war，则可用WinRAR之类的压缩工具打开部署文件smartbi.war并导航到smartbi.war\vision\js\freequery\lang目录下，来获得JSLoader.js。



3、需要修改添加到项目中的JSLoader.js文件。打开该文件，将最后部分的9行代码注释或删除掉，下图中红色矩形框标示的部分。

```
352     }
353     return text.join('');
354 });
355
356     HTMLElement.prototype.__defineSetter__('innerText', function(sText) {
357         this.textContent = sText;
358     });
359 }
360 return jsloader;
361 };
362
363 // set docBase
364 (function(doc) {
365     if (!window.bof_offline_xhr) { //离线的时候不要设docbase
366         var url = (doc.URL.match('[a-zA-Z]+://.+?/vision') || [ '.' ])[0] + '/';
367         var base = doc.createElement('base');
368         base.href = JSLoader.baseHref = url;
369         doc.getElementsByTagName('head')[0].appendChild(base);
370     }
371 }) (document);
```

4、在第三方系统处理用户登录的代码位置（比如登录成功后的代码处理分支处），插入以下单点登录Smartbi的代码。对调用login方法的返回结果进行测试，为true则登录成功；否则登录失败。

```
//smartbi服务器的URL地址
var config = new Object();
config.baseUrl = "http://localhost:8080/smartbi/vision/";
var BOF_UI_DEBUG = false;
// 创建全局唯一的JS装载器
var jsloader = new JSLoader(config);
// 创建应用程序对象
var userService = jsloader.imports("bof.usermanager.UserService");
// 通过userService.getInstance()可以调用所有的UserManagerModule方法.
var result = userService.getInstance().login("admin", "manager");
if (result)
    alert("OK");
```

5、完整的示例代码如下。

```
<HTML>
<HEAD>
  <TITLE>Smartbi </TITLE>
  <SCRIPT type="text/javascript" src="/js/JSLoader.js"></SCRIPT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
  //smartbiURL
  var config = new Object();
  config.baseUrl = "http://localhost:8080/smartbi/vision/";
  var BOF_UI_DEBUG = false;
  // JS
  var jsloader = new JSLoader(config);
  //
  var userService = jsloader.imports("bof.usermanager.UserService");
  // userService.getInstance()UserManagerModule.
  var result = userService.getInstance().login("admin", "manager");
  if (result) {
    alert("OK");
  } else {
    alert("");
  }
</SCRIPT>
</BODY>
</HTML>
```

6、至此单点登录就完成了。但是如果Smartbi与第三方系统部署在不同的机器上，或者部署在同一机器不同的应用服务器域下，即出现跨域访问，则系统在访问过程中有可能出现各种奇怪问题。解决跨域访问，请参考：[Smartbi与第三方系统集成跨域问题解决办法](#)



#### 注意!

JSLoader.js不支持file协议，如果使用该js库的HTML直接在本地打开会报错下面的错误。必须要将其部署到网络上，通过HTTP等协议访问才能正常工作。

报错信息如下：XMLHttpRequest cannot load \${本地URL}. Cross origin requests are only supported for protocol schemes: http, data, chrome, chrome-extension, https.