

从第三方系统中打开Smartbi资源

1. 说明

将报表或页面嵌入到第三方系统中（资源集成），是系统集成中最为常见的一种方式。Smartbi 可以方便地集成到客户现有的门户系统中，或者是其它任意系统中。通常做法是在页面中放置一个 `iframe` 标签，然后通过 `src` 属性访问 Smartbi 的资源链接即可。详述如下。

1.1 资源访问地址

在外部系统中集成 Smartbi 的资源，全部通过下面的 URL 地址来访问。Smartbi 将其所有类型资源的访问全部整合到了 `openresource.jsp` 之中。

```
http://localhost:18080/smartbi/vision/openresource.jsp
```

1.2 支持的资源类型

目前支持被集成的资源类型有：自助仪表盘、多维分析、即席查询、透视分析、Web链接、电子表格、仪表盘页面、可视化查询、SQL数据集、存储过程查询、原生SQL查询，自助数据集以及Excel导入模版等。

1.3 基本参数

下面这几个参数是集成任何类型资源时，都可以传入的参数。但如果已经[单点登录](#)过，就不需要在 URL 中再传递用户名和密码了，除非希望以一个新的用户去打开资源。

资源分类	基本参数
所有资源类型	<ul style="list-style-type: none">• user: 登录 Smartbi 的用户名。• password: 登录 Smartbi 的密码。• resid: Smartbi 资源唯一 ID 值。可以从任意资源的“资源属性”对话框上“节点ID”中获取该值，参见后文说明。• hiddenParamPanel: 是否隐藏参数面板，默认为false。该参数在 v3.0.6 及以后版本上可以使用（自助数据集和自助仪表盘除外）。

传递上述参数，打开某个 Smartbi 资源的访问地址。

从第三方系统中打开Smartbi资源

Smartbi Insight

新标签页

wiki.smartbi.com.cn/pages/viewpage.action?pagelId=44499869

应用 Smartbi 学习 在线网站 idea Spring boot js RBAC SET 维普论文检测【官... set答辩

与第三方系统用户集成

与第三方系统资源集成

从第三方系统链接转到Smartbi

从第三方系统中新建Smartbi资源

从第三方系统中打开Smartbi资源

从第三方系统中打开smartbi模块

集成第三方Java系统并传参数值

从第三方系统中单点登录Smartbi

用第三方系统接管Smartbi相关功能

与第三方GIS系统集成

Smartbi与泛微OA集成示例

钉钉集成Smartbi指南

企业微信扩展包使用流程

自定义任务开发指南

参数控件开发指南

二次开发接口

基本参数

下面这几个参数是集成任何类型资源时，都可以传入的参数。但如果已经单点登录过，就不需要在 URL 中再传递用户名和密码了，除非希望以一个新的用户去打开资源。

资源分类	基本参数
所有资源类型	<ul style="list-style-type: none">• user: 登录 Smartbi 的用户名。• password: 登录 Smartbi 的密码。• resid: Smartbi 资源唯一 ID 值。可以从任意资源的“资源属性”对话框上“节点ID”中获取该值，参见后文说明。• hiddenParamPanel: 是否隐藏参数面板，默认为false。该参数在 v3.0.6 及以后版本上可以使用。

传递上述参数，打开某个 Smartbi 资源的访问地址，如下所示。

```
http://localhost:18080/smartbi/vision/openresource.jsp?resid=I2c949eaf1a942102011a9561f7e7015d&user=admin&password=manager
```

如何获取 Smartbi 资源的 resid 值。打开对应资源的“资源属性”对话框，将其上的“节点ID”拷贝出来即可。

资源属性

节点ID: I402883250162ff83f8384dc01630088254205e5

资源访问地址如下所示。

```
http://localhost:18080/smartbi/vision/openresource.jsp?resid=I2c949eaf1a942102011a9561f7e7015d&user=admin&password=manager
```

如何获取 Smartbi 资源的 resid 值。打开对应资源的“资源属性”对话框，将其上的“节点ID”拷贝出来即可。

- 1. 说明
- 2. 附加参数
- 3. 打开资源示例（不带查询参数）
- 4. 打开资源示例（带查询参数）
- 5. 简易参数传递方式

资源属性

节点ID :

I402883250162ff83ff8384dc01630088254205e5

名字 :

组合分析导出

别名 :

组合分析导出

路径 :

资源定制\V8新特性\组合分析导出

类型名 :

组合分析

描述 :

组合分析导出

创建人 :

管理员

创建时间 :

2018-04-26 14:14:30

显示终端 :

☒ 电脑 ☐ 平板 ☐ 手机

自定义图标 :

上传图片

移动端图标 :

上传图片

移动端缩略图 :

选择文件

未选择任何文件

确定(O)

取消(C)

2. 附加参数

附加参数说明

除了基本参数外，各种资源类型还有一些其它参数可供设置。按照资源类型，分别说明如下。

资源分类	附加参数（可选项）
------	-----------

灵活分析	<p>查询参数：控制灵活分析的数据展示。</p> <ul style="list-style-type: none">paramsInfo: 参数信息，需要对灵活分析的查询参数数组进行JSON序列化。paramsInfoBase64: paramsInfo参数经过Base64编码后的格式，用来解决参数中包含中文和特殊字符问题。与paramsInfo参数，两者选其一。Base64编码是使用sun.misc.BASE64Encoder().encode("你好".getBytes("UTF-8"))的方式转码，如果是直接使用前端处理，需要使用JS的转码方法 <p>显示参数：控制工具栏和是否默认刷新数据等。</p> <ul style="list-style-type: none">showtoolbar: 控制是否在报表的分析页面显示工具栏，默认为true。shorttoolbar: 控制是否在报表的分析页面显示快速工具栏，默认为false不显示。refresh: 打开报表是否刷新数据，默认为false。hidetoolbaritems: 控制报表工具栏中需要隐藏的button，其格式如下： REFRESH SAVE SAVEAS MYFAVORITE <p>中间以空格分割，每个button字符串具体说明如下：</p> <ul style="list-style-type: none"><ul style="list-style-type: none">REFRESH 刷新；SAVE 保存；SAVEAS 另存为；MYFAVORITE 保存到收藏夹；EXPORT 导出；VIEW 视图；SELECTFIELD 增加/删除字段；CHARTSETTING 图形设置；PRINT 打印；PARAMSETTING 参数设置；SUBTOTAL 分类汇总；REPORTSETTING 报表设置。BACKWARD 后退FORWARD 前进CREATEINSIGHTINQUERY 透视分析SETUSERPARAM个人参数shorttoolbaralign: 控制快速工具栏对齐方式，取值如下：<ul style="list-style-type: none">center 居中；left 居左；right 居右 <p>图形参数：用于控制图形类型，比如定制灵活分析时采用的是柱图，集成打开时可改为线图展示。</p> <p>有如下三个参数配套使用，chartindex和chartname必须要传入一个，charttype必须传入。</p> <ul style="list-style-type: none">chartindex: 图形在报表中的位置，string 类型；从左至右、从上至下开始编号，编号从0开始。chartname: 图形名称，string 类型。<ul style="list-style-type: none">如果通过chartname定位图形，必须确保图形没有重名；如果chartindex和chartname都有值，则以chartindex为准定位图形。charttype: 图形类型，string 类型。支持<ul style="list-style-type: none">线图（charttype=Line）、柱图（charttype= StackColumn）、2D饼图（charttype= Pie2D）、油量表（charttype= Gauge）等四种类型。
------	--

即席查询	<p>查询参数：控制灵活分析的数据展示。</p> <ul style="list-style-type: none">• paramsInfo: 参数信息，需要对灵活分析的查询参数数组进行JSON序列化。• paramsInfoBase64: paramsInfo参数经过Base64编码后的格式，用来解决参数中包含中文和特殊字符问题。与paramsInfo参数，两者选其一。 <p>显示参数：控制工具栏和是否默认刷新数据等。</p> <ul style="list-style-type: none">• showtoolbar: 控制是否在报表的分析页面显示工具栏，默认为true。• shorttoolbar: 控制是否在报表的分析页面显示快速工具栏，默认为false不显示。• refresh: 打开报表是否刷新数据，默认为false。• showLeftTree: 集成打开报表是否显示左边的资源树。有如下几种情况：<ul style="list-style-type: none">• 未配置showLeftTree参数时，由系统选项决定是否展示。系统选项路径位于【系统选项】-【高级设置】-【LEFTTREE_DEFAULT_SETTING】<ul style="list-style-type: none">• LEFTTREE_DEFAULT_SETTING=true 表示默认显示• LEFTTREE_DEFAULT_SETTING=false 表示默认不显示• 配置showLeftTree参数时，由showLeftTree参数决定是否显示。<ul style="list-style-type: none">• showLeftTree=true时，显示左侧树。• showLeftTree=false时，不显示左侧树。• showLeftTree=default时，由报表保存时左侧树的显示状态决定。• hidetoolbaritems: 控制报表工具栏中需要隐藏的按钮，其格式如下：REFRESH SAVE SAVEAS MYFAVORITE；中间以空格分割，每个按钮对应的值说明如下：<ul style="list-style-type: none">• REFRESH 刷新；• SAVE 保存；• SAVEAS 另存为；• MYFAVORITE 保存到收藏夹；• EXPORT 导出；• VIEWSQL 查看SQL• VIEW 视图；• CHARTSETTING 图形设置；• PRINT 打印；• PARAMSETTING 参数设置；• REPORTSETTING 报表设置。• FIELDSETTING 字段设置。• CREATEINSIGHTINQUERY 透视分析• MPP 抽取• FILTERRELATIONSSETTING 过滤器关系• USERPARAM 个人参数• TIMECONSUMING 耗时分析
------	---

多维分析	<p>查询参数：控制数据展示。</p> <ul style="list-style-type: none">• paramsInfo: 参数信息，需要对多维分析的查询参数数组进行JSON序列化。• paramsInfoBase64: paramsInfo参数经过Base64编码后的格式，用来解决参数中包含中文和特殊字符问题。与paramsInfo参数，两者选其一。 <p>显示参数：控制工具栏和是否默认刷新数据等。</p> <ul style="list-style-type: none">• refresh: 打开报表是否刷新数据，默认为false。• showtoolbar: 控制是否在报表的分析页面显示工具栏，默认为true。• shorttoolbar: 控制是否在报表分析页面显示快速工具栏，默认为false。• operate: 多维分析的操作类型，取值如下：<ul style="list-style-type: none">• browse 浏览；• discover 探索；• custom 定制。• shorttoolbaralign: 控制快速工具栏对齐方式，取值如下：<ul style="list-style-type: none">• center 居中；• left 居左；• right 居右。• hidetoolbaritems: 控制报表工具栏中需要隐藏的button，其格式如下： REFRESH SAVE SAVEAS MYFAVORITE <p>中间以空格分割，每个button字符串具体说明如下：</p> <ul style="list-style-type: none">• BACKWARD 后退• FORWARD 前进• REFRESH 刷新• SAVE 保存• SAVEAS 另存• MYFAVORITE 添加到收藏夹• CHARTSETTING 图形• PANEL 面板• VIEW 视图• REPORTSETTING 报表设置• ADVANCEDSETTING 高级设置• SUBTOTAL 分类汇总• PARAMSETTING 参数设置• MEMBERSETTING 成员属性设置• CLEARHIDDENSETTINGS 取消隐藏行列• PREVIEWMDX 查看MDX• EXPORT 导出• PRINT 打印
业务查询	<p>显示参数：控制界面元素是否显示。</p> <ul style="list-style-type: none">• hidepropertypanel: 隐藏属性面板。• shorttoolbar: 控制是否显示快速工具栏，默认为false。

透视分析	<p>查询参数：控制数据展示。</p> <ul style="list-style-type: none">paramsInfo: 参数信息，需要对查询参数数组进行JSON序列化。paramsInfoBase64: paramsInfo参数经过Base64编码后的格式，用来解决参数中包含中文和特殊字符问题。与paramsInfo参数，两者选其一。 <p>字段设置参数：控制行区列区的字段。</p> <ul style="list-style-type: none">fieldsConfig: 放行区列区度量区的字段信息数组，需要对字段信息数组进行JSON序列化。字段的内容包括： <pre>{ name : "字段名，和字段别名只需提供任意一个", alias : "字段别名，和字段名只需提供任意一个", pos : "row/column/measure中的一个，指定字段所在的区域", num : 字段在行区/列区/度量区中的顺序号 }</pre> <p>显示参数：控制工具栏和是否默认刷新数据等。</p> <ul style="list-style-type: none">refresh: 打开报表是否刷新数据。如果不指定，以报表保存时的设置为准。showtoolbar: 控制是否显示工具栏，默认为true。showWorkspace: 控制是否显示右侧的工作区面板，默认为false。showFieldPopupMenu: 控制是否在右侧的工作区面板中显示右键菜单，默认为true。showResource: 控制是否显示左侧的业务主题面板，默认为false。hidetoolbaritems: 控制报表工具栏中需要隐藏的按钮，其格式如下：REFRESH SAVE SAVEAS MYFAVORITE；中间以空格分割，每个按钮对应的值说明如下：<ul style="list-style-type: none">SAVE 保存；SAVEAS 另存为；BTNPARAMSETTING 参数设置PANEL 面板REPORTSETTING 报表设置WARNING 告警REFRESH 刷新MYFAVORITE 添加到收藏夹CHARTSETTING 图形VIEW 视图EXPORT 导出PRINT 打印SETUSERPARAM 个人参数RESOURCE TREE 资源树FILTERRELATION 过滤TIMECONSUMING 耗时分析
电子表格	<p>查询参数：控制数据展示。</p> <ul style="list-style-type: none">paramsInfo: 参数信息，需要对多维分析的查询参数数组进行JSON序列化。paramsInfoBase64: paramsInfo参数经过Base64编码后的格式，用来解决参数中包含中文和特殊字符问题。与paramsInfo参数，两者选其一。 <p>显示参数：控制工具栏和是否默认刷新数据等。</p> <ul style="list-style-type: none">showtoolbar: 控制是否在报表的分析页面显示工具栏，默认为true。shorttoolbar: 控制是否在报表分析页面显示快速工具栏，默认为false。hiddenParamPanel: 控制是否在报表分析页面显示参数面板，默认为true。isWritable: 控制回写报表是否允许回写。hidetoolbaritems: 控制报表工具栏中需要隐藏的按钮，其格式如下：REFRESH MYFAVORITE；中间以空格分割，每个按钮对应的值说明如下：<ul style="list-style-type: none">REFRESH 刷新；MYFAVORITE 保存到收藏夹；EXPORT 导出；PRINT 打印。SETUSERPARAM 个人参数
自助仪表盘	<p>查询参数：控制数据展示。</p> <ul style="list-style-type: none">paramsInfo: 参数信息，需要对查询参数对象(key为筛选器组件的标题)进行JSON序列化。 参数对象的属性: values, 参数值。 示例: openresource.jsp?resid=e06fd037ef41336fe79908f60e961c95&paramsInfo={"OrderDate":{"values":["~2016-01-03"]}}paramsInfoBase64: paramsInfo参数经过Base64编码后的格式，用来解决参数中包含中文和特殊字符问题；与paramsInfo参数，两者选其一。 示例: openresource.jsp?resid=e06fd037ef41336fe79908f60e961c95&paramsInfoBase64=eyJPcmRlckRh dGUiO nsidmFsdWVzIjpbIjIwMTYtMDEtMDMxX19 <p>显示参数</p> <ul style="list-style-type: none">operate: 操作方式，目前仅支持operate=edit或operate=EDIT，可选参数，传递该参数代表编辑自助仪表盘

Excel导入模版	显示参数: 控制显示模式 <ul style="list-style-type: none">type: 操作方式, 仅支持type=edit或type=EDIT, 可选参数, 传递该参数代表编辑Excel导入模版
-----------	--

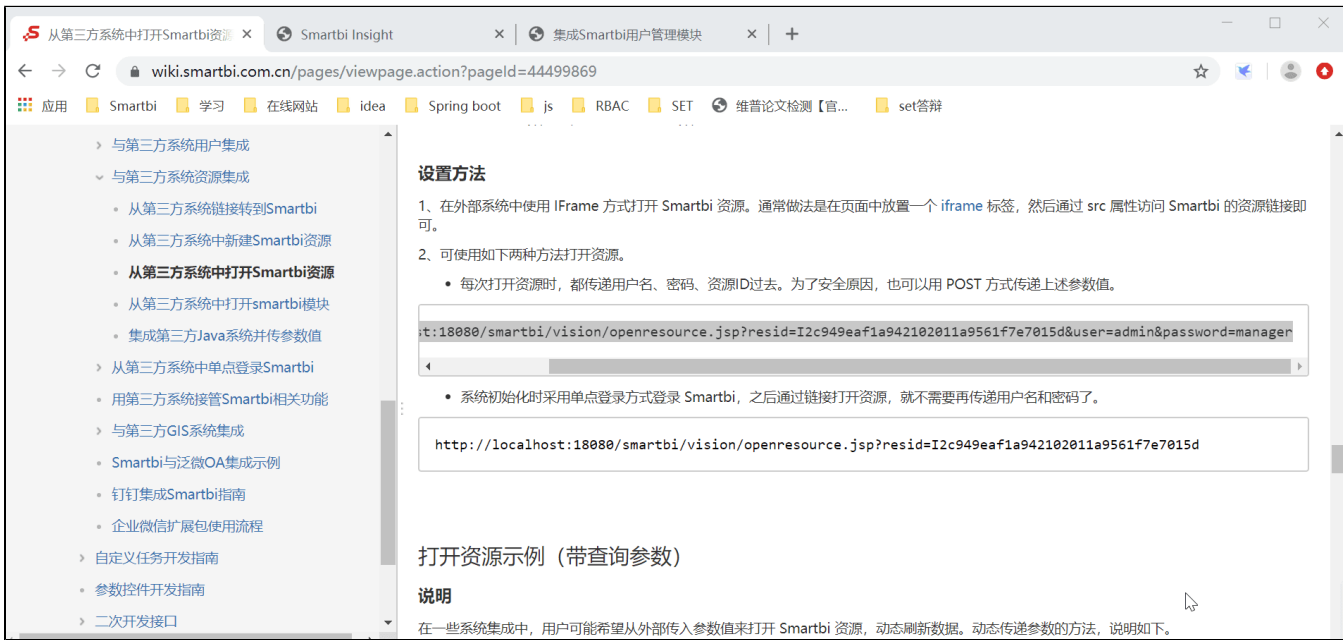
注意: base64后的字符串需要encode一下, 否则存在特殊字符时会乱码。参考写法:

```
<%
    String url = "http://localhost.:15000/smartbi/vision/openresource.jsp?
resid=I2c949e8e1ac2d5e6011ac380971301b8";
    JSONArray paramArray = new JSONArray();
    JSONObject paramObject = new JSONObject();
    paramObject.put("name", "");
    paramObject.put("value", "");
    paramObject.put("displayValue", "");
    paramArray.put(paramObject);
    String paramString = paramArray.toString();
    String base64ParamString = java.net.URLEncoder.encode((new sun.misc.BASE64Encoder()).encode
(paramString.getBytes("UTF-8")));
    url += "&paramsInfoBase64=" + base64ParamString;
    url += "&refresh=true";
    //System.out.println("url: " + url);
    response.sendRedirect(url);
%>
```

3. 打开资源示例（不带查询参数）

3.1说明

下面仅介绍基本参数传递, 不带任何附加参数。



3.2设置方法

1、在外部系统中使用 IFrame 方式打开 Smartbi 资源。通常做法是在页面中放置一个 `iframe` 标签, 然后通过 `src` 属性访问 Smartbi 的资源链接即可。

2、可使用如下两种方法打开资源。

方式一: 每次打开资源时, 都传递用户名、密码、资源ID过去。为了安全原因, 也可以用 POST 方式传递上述参数值。

```
http://localhost:18080/smartbi/vision/openresource.jsp?
resid=I2c949eaf1a942102011a9561f7e7015d&user=admin&password=manager
```

方式二：系统初始化时采用单点登录方式登录 Smartbi，之后通过链接打开资源，就不需要再传递用户名和密码了。

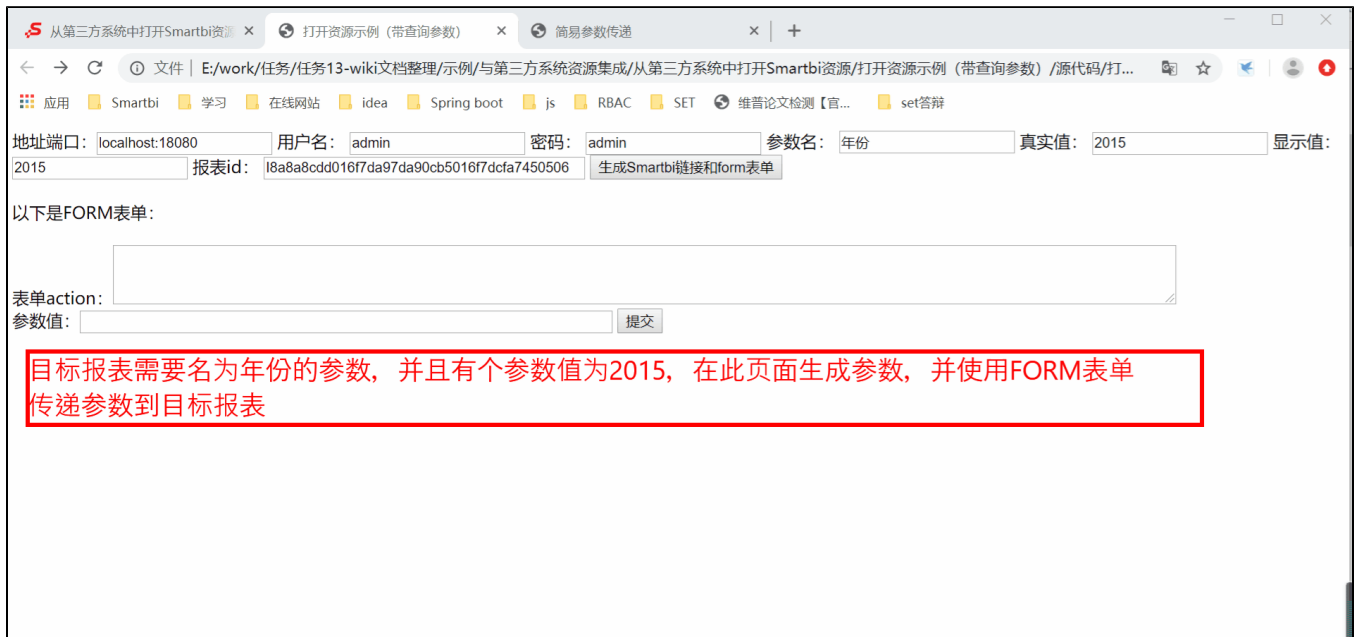
```
http://localhost:18080/smartbi/vision/openresource.jsp?resid=I2c949eaf1a942102011a9561f7e7015d
```

3、gif演示示例源码请参考：[打开资源示例（不带查询参数）.rar](#)

4. 打开资源示例（带查询参数）

4.1 说明

在一些系统集成中，用户可能希望从外部传入参数值来打开 Smartbi 资源，动态刷新数据。动态传递参数的方法，说明如下。



4.2 设置步骤

1、将下面这段 JavaScript 代码拷贝到第三方系统的某个页面里，因为后面我们需要使用其中的 **function toJSONString(obj)** 方法对传递的参数值进行处理。

```
function parseJSON (jsonString) { //FROM: json.org,json.js
    return eval('(' + jsonString + ')');
};

function toJSONString (obj) { //FROM: json.org,json.js
    var m = {
        '\b': '\\b',
        '\t': '\\t',
        '\n': '\\n',
        '\f': '\\f',
        '\r': '\\r',
        '"' : '\\"',
        '\\': '\\\\'
    };
    var s = {
        array: function (x) {
            var a = ['[', b, f, i, l = x.length, v;
            if(l == 0) {
                var isNull = true;
                for(var i in x) {
                    // Fenet's code begin
                    if ( i == "remove" || i == "indexOf" ) {
```



```

        delete x[i];
        continue;
    }
    if( x.toJSON)
        return x.toJSON();
    // Fenet's code end
    isNull = false;
    break;
}
if(!isNull) {
    //throw null;
    var a = ['{'], b, f, i, v;
    for (i in x) {
        v = x[i];
        f = s[typeof v];
        if (f) {
            v = f(v);
            if (typeof v == 'string') {
                if (b) {
                    a[a.length] = ',';
                }
                a.push(s.string(i), ': ', v);
                b = true;
            }
        }
    }
    a[a.length] = '}';
    return a.join('');
} else {
    for (i = 0; i < l; i += 1) {
        v = x[i];
        f = s[typeof v];
        if (f) {
            v = f(v);
            if (typeof v == 'string') {
                if (b) {
                    a[a.length] = ',';
                }
                a[a.length] = v;
                b = true;
            }
        }
    }
}
a[a.length] = ']';
return a.join('');
},
'boolean': function (x) {
    return String(x);
},
'null': function (x) {
    return "null";
},
number: function (x) {
    return isFinite(x) ? String(x) : 'null';
},
object: function (x) {
    if (x) {
        if (Object.prototype.toString.call(x) === "[object Array]") {
            return s.array(x);
        }
        var a = ['{'], b, f, i, v;
        for (i in x) {
            v = x[i];
            f = s[typeof v];
            if (f) {
                v = f(v);
                if (typeof v == 'string') {
                    if (b) {
                        a[a.length] = ',';
                    }
                }
            }
        }
    }
}

```

```

    }
    a.push(s.string(i), ':' + v);
    b = true;
  }
}
a[a.length] = '';
return a.join('');
}
return 'null';
},
string: function (x) {
  if (/["\\x00-\x1f]/.test(x)) {
    x = x.replace(/(["\\x00-\x1f])/g, function(a, b) {
      var c = m[b];
      if (c) {
        return c;
      }
      c = b.charCodeAt();
      return '\\u00' + Math.floor(c / 16).toString(16) +
(c % 16).toString(16);
    });
  }
  return '"' + x + '"';
}
};
return s[typeof obj](obj);
};

```

2、构建参数信息对象数组。每个参数都是一个 JavaScript 对象，可设置如下属性。

属性	名称	类型	备注
id	参数名称	String类型	多维分析必选。
name	参数名称	String类型	必选，多维分析可不选用。
value	参数真实值	String类型	必选。参数真实值可在参数定义中查看。
displayValue	参数显示值	String类型	必选。
stanbyValue	下拉列表参数的备选值	String类型	可选项。当参数类型是下拉列表框时，可选择设置该属性值。 <ul style="list-style-type: none"> 如果不设置，则会使用原有的参数设置； 如果设置，系统中原有的参数备选值就会被忽略。 用法："stanbyValue":[[华南,华南],[华北,华北]]，注：后面数组值不需要用引号引起来

3、将参数列表对象转换成 JSON 字符串。通过调用步骤1中的 toJSONString() 函数可以将 JavaScript 对象序列化化成 json 字符串，将此部分代码写在集成页面中。注意下图红色矩形框内代码写法，如果一次传递多个参数就向 paramsInfo 数组中添加多个对象。

```

//生成待传递的参数数组对象
var paramProductName = {
  "name": paramName.value,
  "value": realValue.value,
  "displayValue": displayValue.value
};
var paramsInfo = [];
paramsInfo.push(paramProductName);

//生成form表单的action属性并赋值
var urlStr = "http://" + address.value + "/smartbi/vision/openresource.jsp"
+ "?resid=" + resid.value
+ "&showtoolbar=true"
+ "&refresh=true"
+ "&user=" + user.value
+ "&password=" + password.value;

```

```
//Smartbi
function generate(){
    //
    var paramProductName = {
        "name": paramName.value,
        "value": realValue.value,
        "displayValue": displayValue.value
    };
    var paramsInfo = [];
    paramsInfo.push(paramProductName);

    //formation
    var urlStr = "http://" + address.value + "/smartbi/vision/openresource.jsp"
        + "?resid=" + resid.value
        + "&showtoolbar=true"
        + "&refresh=true"
        + "&user=" + user.value
        + "&password=" + password.value;
    url.value = urlStr + "\nformparamsInfo=" + toJSONString(paramsInfo);
    form.action = urlStr;

    //form
    //
    paramsInfoForm.value = toJSONString(paramsInfo);
}
}
```

4、将user、password、resid、paramsInfo 等信息传递到 openresource.jsp 地址，即可打开相关资源。如下所示。

```
http://localhost:18080/smartbi/vision/openresource.jsp?paramsInfo=[{"name":"","value":"","displayValue":""}]
&resid=I2c949e8elac2d5e6011ac380971301b8&showtoolbar=true&refresh=true&user=admin&password=manager
```

5、gif演示示例的源码请参考：[打开资源示例（带查询参数）.rar](#)



注意：

- 上述这种将 paramsInfo 放在URL链接中的方式，一定要将其放到 ? 后面作为第一个参数。否则在某些浏览器上会出错，&p 被显示成了一个乱码。
- 另外需要用 encodeURIComponent() 函数对 paramsInfo 进行编码，对其中的中文、特殊符号进行处理。Smartbi 小工具用来生成 paramsInfo 并编码：<http://localhost:18080/smartbi/vision/getparamsinfo.html>
- 这种将参数直接放到URL中的方式，在实际项目上不太合适的，一方面 URL 长度有限制；再者如果参数中有中文和特殊符号时，就很容易出错。因此建议：最好通过 POST 的方式将参数值传递到 URL 地址上。
- 在Tomcat高版本部署smartbi时，如果请求的URL在编码后的中文字符串带反斜杠，根据RFC文档中规定反斜杠是不安全字符，Tomcat在高版本中增加的安全验证，凡是RFC 3986中非URL可携带的字符，都会返回400错误。
- 第三方系统集成smartbi的资源需要注意跨域问题。
- 本示例不适用于自助仪表盘。

5. 简易参数传递方式

5.1说明

paramsInfo 在实际项目中使用时稍显麻烦，需要构造参数对象数组，并生成 json 字符串。因此在 Smartbi 2.5 以后版本上提供了新的简易参数传递方式，以**param** 作前缀表示参数实际值，**paramDisplay** 前缀作为参数显示值。

从第三方系统中打开Smartbi资源 x 简易参数传递 x +

文件 | E:/work/任务/任务13-wiki文档整理/示例/与第三方系统资源集成/从第三方系统中打开Smartbi资源/简易参数传递方式/源代码/简易参数传递.h... ☆

应用 Smartbi 学习 在线网站 idea Spring boot js RBAC SET 维普论文检测【官... set答疑

地址端口: localhost:18080 用户名: admin 密码: admin 参数名: 年份 真实值: 2015

显示值: 2015 报表id: l8a8a8cdd016f7da97da90cb5016f7dcfa7450506 生成Smartbi链接和form表单

以下是FORM表单:

表单action: 真实值: 显示值: 提交

目标报表需要名为年份的参数,并且有个参数值为2015,在此页面生成参数,并使用FORM表单传递参数到目标报表, 参数为param.年份=2015, paramDisplay.年份=2015

5.2 设置步骤

1、后台表单如下所示。

```
<form action="http://localhost:18080/smartbi/vision/openresource.jsp?
resid=I2c90903e114f6f9601114f70e09d000e&refresh=true" method="POST">
  <input type="text" name="param." value="1996-03-17">
  <input type="text" name="paramDisplay." value="1996-03-17">
  <input type="text" name="param." value="1996-09-19">
  <input type="text" name="paramDisplay." value="1996-09-19">
  <input type="text" name="param." value="8">
  <input type="text" name="paramDisplay." value="">
  <input type="submit">
</form>
```

2、gif演示示例的源码请参考: [简易参数传递方式.rar](#)

这样就比较简单了,也无需在项目中使用JSON作为类库,调用其中的toJSONString()方法了。但是需要注意,如果在URL中直接传递参数,参数值和名称等存在中文时容易出现问題。最好使用POST方式,或者在URL中以UTF-8编码传递,并修改tomcat/conf/server.xml中Connector中添加URIEncoding="UTF-8"属性。