从第三方系统中打开Smartbi资源

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。

文档目录：

- 1、概述
  - 1.1 资源访问地址
  - 1.2 支持的资源类型
- 2、基本参数
- 3、附加参数
  - 3.1 附加参数说明
- 4、打开资源示例（不带查询参数）
  - 4.1 说明
  - 4.2 设置方法
- 5、打开资源示例（带查询参数）
  - 5.1 说明
  - 5.2 设置步骤
  - 5.3 简易参数传递方式

1、概述

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、组合分析、透视分析、Web链接、电子表格、可视化查询、SQL查询、存储过程查询、原生SQL查询，自助数据集以及Excel导入模版等。

2、基本参数

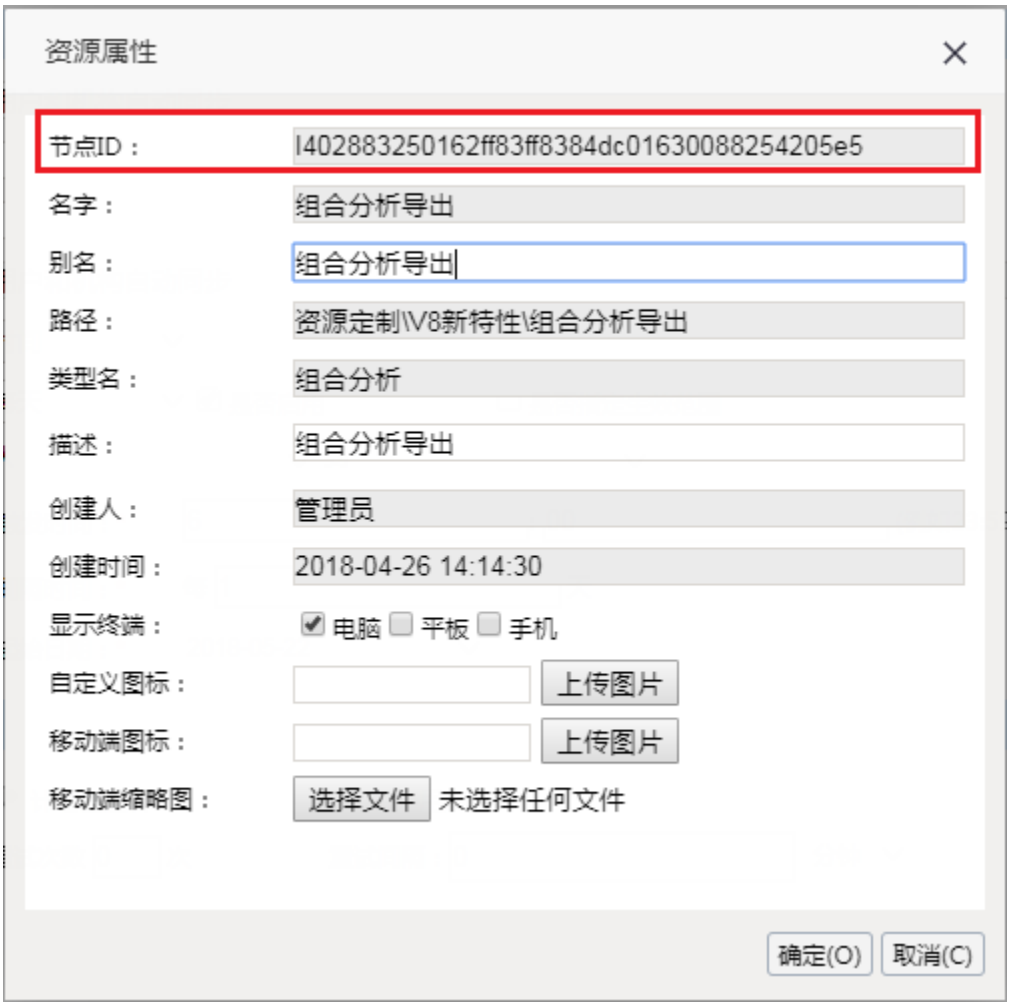
- 下面这几个参数是集成任何类型资源时，都可以传入的参数。但如果已经单点登录过，就不需要在 URL 中再传递用户名和密码了，除非希望以一个新的用户去打开资源。

资源分类	基本参数
所有资源类型	• 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，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。



3、附加参数

3.1 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	**查询参数：**控制灵活分析的数据展示。
	
	• 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。该参数与paramsInfo参数，两者只能选其一不可同时使用。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，需要自己转码方法。
	
	显示参数：控制工具栏和是否默认刷新数据等。
	
	• 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 中间以空格分割，每个button字符串具体说明如下：
	  • 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：
	  • center 居中；
	  • left 居左；
	  • right 居右；
	
	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。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。
	
	• 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。
	  • 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。支持如下值：
	  • 线图（charttype=Line）；
	  • 柱图（charttype= StackColumn）；
	  • 2D饼图（charttype= Pie2D）；
	  • 油量表（charttype= Gauge）；

组合分析	**查询参数：**控制灵活分析的数据展示。
	
	• 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。该参数与paramsInfo参数，两者只能选其一。
	
	显示参数：控制工具栏和是否默认刷新数据等。
	
	• 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：打开报表是否显示左边的资源树，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 中间以空格分割，每个按钮对应的值说明如下：
	  • 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；
	  • SETUSERPARAM 个人参数；
	  • REPORTSETTING 报表设置；
	  • FIELDSETTING 字段设置；
	  • CREATEINSIGHTINQUERY 透视分析；
	  • CONDITIONSETTING 条件；
	  • MPP 数据抽取；

多维分析	**查询参数**：控制数据展示。
	
	• 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。该参数与paramsInfo参数，两者只能选其一。
	
	显示参数：控制工具栏和是否默认刷新数据等。
	
	• 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：
	  • 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：
	  • 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 中间以空格分割，每个button字符串具体说明如下：
	
	  • 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	**显示参数**：控制界面元素是否显示。 • 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	**查询参数：**控制数据展示。
	
	• 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。该参数与paramsInfo参数，两者只能选其一。
	
	字段设置参数：控制行区列区的字段。
	
	• **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：
	
	{
	
	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
	
	}
	
	显示参数：控制工具栏和是否默认刷新数据等。
	
	• **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 中间以空格分割，每个按钮对应的值说明如下：
	  • 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；
	  • RESOURCETREE 资源树；
	  • FILTERRELATION 过滤；

电子表格	**查询参数**：控制数据展示。
	
	• 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。该参数与paramsInfo参数，两者只能选其一。
	
	显示参数：控制工具栏和是否默认刷新数据等。
	
	• 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版，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：
	  • 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	**查询参数**：控制数据展示。 • 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；该参数与paramsInfo参数，两者只能选其一。 示例：openresource.jsp? resid=e06fd037ef41336fe79908f60e961c95&paramsInfoBase64=eyJPcmRlckRhdGUiOnsidmFsdWVzIjpbIjIwMTYtMDEtMDMiXX19 **显示参数** • 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

提示：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);
%>
```

4、打开资源示例（不带查询参数）

4.1 说明

下面仅介绍基本参数传递，不带任何附加参数。

4.2 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。

- 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
```

- 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
```

5、打开资源示例（带查询参数）

5.1 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。

5.2 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。 • 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 对象序列化成 json 字符串，将此部分代码写在集成页面中。注意下图中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。

名称：报表跳转_action　　类型：ClientSide　　对象：simpleReport　　事件：onRender

```
function main(simpleReport, simpleReportContext) {
    simpleReport.addColumnLinkByName("ProductName", menuItem);
};
var menuItem = {
    alias: "报表跳转",
    callback: function(simpleReport, simpleReportContext) {
        //获取当前单元格所在行的产品名称字段的值
        simpleReport.writeParamsToContext();
        var realValue = simpleReport.getCurrentFieldRealValue("产品名称");
        var dispValue = simpleReport.getCurrentFieldValue("产品名称");
        //生成待传递的参数数组对象
        var paramProductName = {
            "name": "产品名称参数",
            "value": realValue,
            "displayValue": dispValue
        };
        var paramsInfo = [];
        paramsInfo.push(paramProductName);
        //paramsInfo.push(paramProductClass);
        //paramsInfo.push(paramOther);
        var url = "openresource.jsp"
                + "?paramsInfo=" + lang.toJSONString(paramsInfo)
                + "&resid=I2c949e8e1ac2d5e6011ac380971301b8"
                + "&showtoolbar=true"
                + "&refresh=true";
        simpleReportContext.openURL(url); //在新窗口打开资源
    }
};
```

```
function main(simpleReport, simpleReportContext) {
    simpleReport.addColumnLinkByName("ProductName", menuItem);
};
var menuItem = {
    alias: "",
    callback: function(simpleReport, simpleReportContext) {
        //
        simpleReport.writeParamsToContext();
        var realValue = simpleReport.getCurrentFieldRealValue("");
        var dispValue = simpleReport.getCurrentFieldValue("");
        //
        var paramProductName = {
            "name": "",
            "value": realValue,
            "displayValue": dispValue
        };
        var paramsInfo = [];
        paramsInfo.push(paramProductName);
        //paramsInfo.push(paramProductClass);
        //paramsInfo.push(paramOther);
        var url = "openresource.jsp"
                + "?paramsInfo=" + lang.toJSONString(paramsInfo)
                + "&resid=I2c949e8e1ac2d5e6011ac380971301b8"
                + "&showtoolbar=true"
                + "&refresh=true";
        simpleReportContext.openURL(url); //
    }
};
```

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=I2c949e8e1ac2d5e6011ac380971301b8&showtoolbar=true&refresh=true&user=admin&password=manager
```

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。
- 另外需要对其中的中文转成unicode，然后用 encodeURIComponent() 函数对paramsInfo里面的特殊符号进行编码。Smartbi 小工具用来生成 paramsInfo 并编码：http://localhost:18080/smartbi/vision/getparamsinfo.html，以下为生成的示例：http://localhost:8080/smartbi/vision/openresource.jsp?resid=I8a8a8c1301759168916803af01759210a68805f3&paramsInfo=%5B%7B%22name%22:%22input%22,value:%22%5Cu996e%5Cu6599%22%7D%5D
- 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，详细说明文档请看：Tomcat高版本部署smartbi通过URL拼接参数打开报表报400

5.3 简易参数传递方式

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。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>
```

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题。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。