

Smartbi前端框架介绍

1. Smartbi 前端框架

1.1前端组件框架

Smartbi 是典型的基于JavaScript的面向对象框架，整个系统只有几个入口jsp（譬如index.jsp、login.jsp），剩下基于AJAX按业务或操作逻辑按需动态加载或注销组件，譬如在系统中双击一张分析报表，系统就会调出报表组件（QueryView）并执行打开的操作。下图是简单的前端组件图，对于定制来说最常做的操作是编写或修改业务逻辑层的组件。

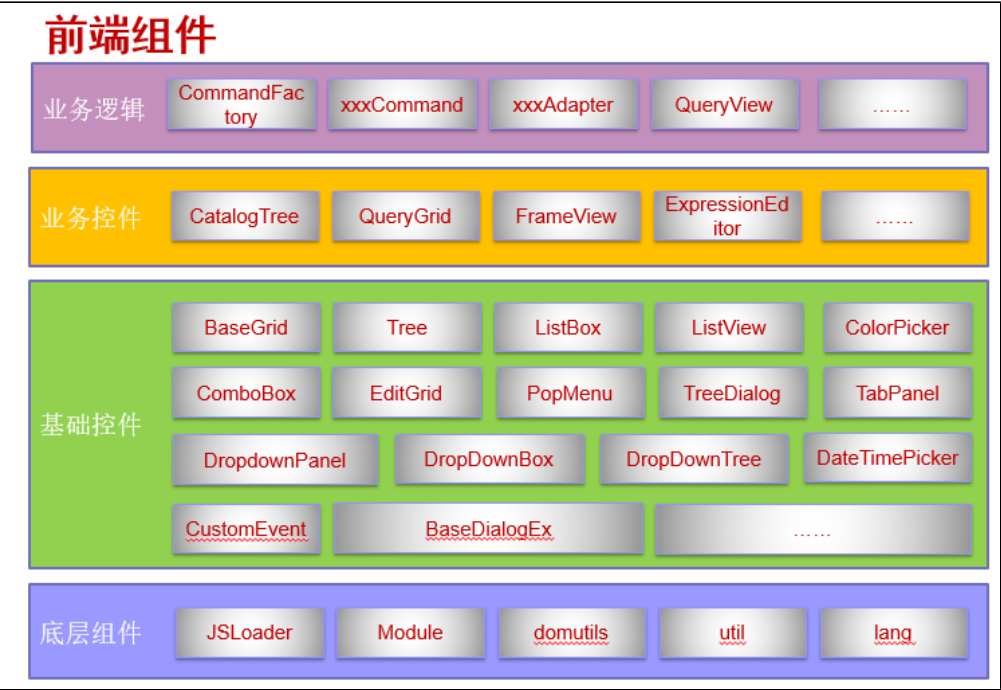
下图分为四层：

- 1、**底层组件**：是工具类、抽象接口类性质，下面章节会重点介绍几个常用的。
- 2、**基础控件**：类似下拉框、表格、树、tab、列表、弹窗等界面控件或对象。

- 3、**业务控件**：基于基础控件又封装的一层具有业务意义的控件，譬如定制管理左侧的资源树就是资源树控件。

4、**业务逻辑**：整个系统只有几个入口jsp，并不是说所有内容一次性加载，而是根据用户鼠标操作按需加载或注销内容，所以系统的每个功能，其实都会对应一个js组件，譬如电子表格、透视分析、多维分析等都会有自己对应的组件，通过定制给系统增加一个功能界面也相当于要创建一个业务逻辑组件。

- 1. Smartbi 前端框架
 - 1.1前端组件框架
 - 1.2前后端通信框架
- 2. 关键组件介绍
 - 2.1JSLoader：提供加载js的方法
 - 2.2domutils：dom元素工具类
 - 2.3util：前后端交互工具类
 - 2.4lang：js集成和重写工具类
 - 2.5CustomEvent：自定义事件
 - 2.7BaseDialogEx：对话框组件基类



1.2前后端通信框架

这里分两类介绍前后端通信，文件类交互请求，譬如js、html、css，和操作或数据交互类请求，譬如刷新报表，新建报表之类。

1、文件类交互：

(1) JS文件：使用 jsloader 方式按需加载js文件，jsloader 也是封装了请求gbk.jsp?name=jsname，见“[第四课：如何修改Smartbi JS文件](#)”里的说明；

(2) CSS文件：为了减少css的请求，系统采用bof_merge.css.jsp将css合并一次性加载，同时提供了扩展点修改系统内的样式或新增样式，扩展点中配置的css文件，bof_merge.css.jsp会自动识别并加载；

(3) HTML文件：html文件一般作为组件的布局和展现文件，下面介绍的“Module2：js组件基类”，如果使用了基类的init方法，默认会自动加载同名的.html文件，也可以使用domutils.doGet(“相对于vision的html完整路径”)方式加载指定的.html或.template文件，例如 var html = domutils.doGet(“template/frequery/query/QueryView.template”);

2、**操作和数据交互**：使用util.remoteInvokeEx/remoteInvoke与服务端module实现数据交互，如果是希望与jsp/servlet交互，可以使用domutils中提供的doPost/doGet方法交互。很多时候写js组件时，可能都会有与服务端交互的场景，这时候也是需要编写自定义module的，然后在js组件中使用util.remoteInvokeEx/remoteInvoke调用这个module。

2. 关键组件介绍

2.1JSLoader：提供加载js的方法

Smartbi 内置了一个全局对象jsloader，就是通过这个脚本对象提供的几个方法异步加载js，如果很多地方重复加载一个js，系统会自动缓存，只会从服务器请求一次。jsloader本质是使用gbk.jsp加载。要能让jsloader正常加载到js，需要遵循：

- 1、所有js文件必须置于“vision/js/”目录或其子目录下。
- 2、js文件名(不含后缀)及其所有父目录的文件名都不能包括“.”字符。
- 3、模块内必须定义一个与文件名同名的全局变量。

jsloader 中三个加载js方法说明见下文：

1、resolve(name, useGlobal)：按需加载并执行

参数说明：

- (1) **name**：要加载的js名称，是相对于vision/js目录的完整路径名，譬如加载vision/js/freequery/lang/CustomEvent，就是“freequery.lang.CustomEvent”
- (2) **useGlobal**：是否全局，缺省为false，如果为true，就等同于使用<script>标签加载

resolve示例

```
// index.jsp<script>
jsloader.resolve('thirdparty.jquery.jquery', true);

// resolve
var util=jsloader.resolve("freequery.common.util");
var CustomEvent = jsloader.resolve("freequery.lang.CustomEvent"); //freequery.lang.CustomEvent
```

2、resolveMany(names)：批量加载但不执行，意思是批量从服务端请求js，减少与服务端沟通时间

参数说明

- (1) **names**：要加载的js名称数组，是相对于vision/js目录的完整路径名，如['freequery.common.util','bof.baseajax.common.Application']

resolveMany示例

```
jsloader.resolveMany(['freequery.common.util','bof.baseajax.common.Application','freequery.widget.Module' ]);

//
var util=jsloader.resolve("freequery.common.util"); //eval
```

3、imports(className)：仅声明待用到再加载与执行

参数说明

- (1) **className**：要加载的js名称，是相对于vision/js目录的完整路径名，譬如：“freequery.lang.CustomEvent”

imports示例

```
var util = jsloader.imports("freequery.common.util");
// getInstance() util.getInstance().remoteInvokeEx(...)
```

2.2domutils：dom元素工具类

工具类，提供判断浏览器版本、异步请求、给dom元素增加css类等工具类方法，详细可以查看smartbi.war/vision/js/freequery/lang/domutils.js，这里列出几个常用的方法：

1、doGet(url,notUseGBKJSP)：使用get方式请求

参数说明：

- (1) **url**：要请求的url地址，如果是系统内部资源，是相对于vision的url，譬如：template/freequery/query/QueryView.template
- (2) **notUseGBKJSP**：是否使用gbk.jsp加载，默认为false，gbk.jsp是系统用于加载js,.template,.html文件的一个jsp

doGet示例

```
var template = domutils.doGet("template/freequery/query/QueryView.template");
this.panel = document.createElement("div");
this.panel.innerHTML = template;
```

2. doPost(url, data, callback, errorHandler, scope, headers): 使用post方式请求

参数说明:

- (1) **url**: 如果是系统资源, 相对于vision地址的url
- (2) **data**: post的数据, 譬如: "A=xx&B=yy"
- (3) **callback**: 请求成功返回的回调函数, 如果传递了此方法就是异步请求, 否则同步请求
- (4) **errorHandler**: 请求异常的回调函数, 只有传递了callback参数时, 此参数才生效
- (5) **scope**: callback函数内部的this对象
- (6) **headers**: 请求头信息, json对象, 譬如{If-Modified-Since:0}

doPost示例片段

```
// doPost
var url = "RMIServlet"; //url
var data = null; //
data = "className=" + encodeURIComponent(className) +
      "&methodName=" + encodeURIComponent(methodName) +
      "&params=" + encodeURIComponent(paramsStr);

domutils.doPost(url, data, function(responseText) {
    var export2FtpUtil = jsloader.resolve("aladdin.utils.Export2FtpUtil");
    export2FtpUtil.showExportResult(responseText);
}, function(xhr) {
    alert("");
}, this, null);
```

3. addClassName/removeClassName/hasClassName(elem, value): 给dom元素添加或删除css样式类

参数说明:

- (1) **elem**: dom元素对象
- (2) **value**: 样式类名

```
if (domutils.hasClassName(elem, 'awesomplete')) {
    domutils.addClassName(elem, 'search-wrapper'); //domutils.removeClassName(elem, 'search-wrapper');
}
```

4. isIE(): 是否是IE

is+浏览器英文名代表判断是否xx浏览器的方法, 例如isFirefox、isIE、isIE6、isIE11、isEdge、isChrome、isQQBrowser、isSafari、isOpera、isIE7、isIOS、isAndroid、isMobile。

2.3util: 前后端交互工具类

这个工具类最关键的一个功能是提供了客户端与服务端module直接沟通的方法, 详细的方法可以查看smartbi.war/vision/js/freequery/common/util.js, 这里只介绍几个关键方法:

1、remoteInvokeEx /remoteInvoke(className, methodName, paramArray, callback, that, headers): 与服务端module沟通方法, 其中remoteInvokeEx如果同步请求出现异常会自动弹窗提示。

参数说明:

- (1) **className**: 配置在applicationContext.xml中注册到rmi中的名称, 譬如下面示例中就是ExtSample8Service
 - (2) **methodName**: 要请求module中的哪个方法
 - (3) **paramArray**: 上面方法接收的参数数组, 数组中的第一个对应方法的第一个参数, 依次类推
 - (4) **callback**: 回调函数, 请求返回执行, 如果不传递此参数代表同步请求
 - (5) **that**: callback里的this对象
 - (6) **headers**: 请求头信息, 譬如: json对象, 譬如{If-Modified-Since:0}
- 可执行示例请见 [宏代码中执行sql语句](#)。

module调用示例

```
//
//
var ret = util.remoteInvoke("DashboardService", "getParamValueFromDashboard", [this.clientId, paramId]);
if (ret.succeeded) {
    return ret.result;
} else {
    modalWindow.showServerError(ret);
}

//
var ret = util.remoteInvoke("DashboardService", "getParamValueFromDashboard", [this.clientId, paramId],
function(ret){
    if(ret.succeeded){
        var result = ret.result; //getParamValueFromDashboardjson
    }
}, this);
```

- 2、getCookie(name)：获取指定名称cookie值。
- 3、getSystemConfig(key)：获取指定key的系统选项值。

2.4lang: js集成和重写工具类

提供了类的继承方法、重写等功能。

- 1、extend (subclass, superclass)：类的继承

继承示例

```
var BaseDialogEx = jsloader.resolve("freequery.dialog.BaseDialogEx");

var ExportResultDialog = function() {
    //
};
lang.extend(ExportResultDialog, BaseDialogEx);

ExportResultDialog.prototype.init = function(parent, data, fn, obj) {
    ExportResultDialog.superclass.init.call(this, parent, data, fn, obj);
    //BaseDialogEx.superclass.init.call(this, this.dialogBody, __url, true);
    var cookie = document.cookie;
    var cookieAry = cookie.split(';');
    var downloadSrcCookie;
    for (var i in cookieAry) {
        if (cookieAry[i].indexOf('download_src') != -1) {
            downloadSrcCookie = cookieAry[i];
            break;
        }
    }
    var info = "";
    this.dialogBody.style.paddingTop = "30px";
    if (downloadSrcCookie) {
        var src = downloadSrcCookie.substring(downloadSrcCookie.indexOf('=') + 1);
        linkStr = '<a target="_blank" href="http://ip:8080/secdoc/encrypt?file=' + src + '"></a>';
        this.dialogBody.innerHTML = info + linkStr;
    } else {
        this.dialogBody.innerHTML = info;
    }
}

ExportResultDialog.prototype.destroy = function() {
    ExportResultDialog.superclass.destroy.call(this);
}
```

- 2、patch (subclass, superclass)：提供重写js类的构造方法，请见 [如何修改Smartbi JS文件](#)。
- 3、parseJSON (jsonString)
- 4、toJSONString(obj)

toJSONString和parseJSON调用示例

```
jsloader.resolve("freequery.lang.lang");
var testObj = {
    name:"test",
    age:"21"
};
var resultStr = lang.toJSONString(testObj);
testObj = lang.parseJSON(resultStr);
```

2.5CustomEvent：自定义事件

组件希望在某个点抛出事件，供外面调用组件的地方注册使用时，可以使用这个类，例如电子表格刷新完成抛出onAfterRefresh事件：

CustomEvent应用示例

```
var util = jsloader.resolve('freequery.common.util');
var CustomEvent = resolve("freequery.lang.CustomEvent");

var SpreadsheetReport = function(container) {
    SpreadsheetReport.superclass.constructor.call(this, container);
    //
    this.onAfterRefresh = new CustomEvent("AfterRefresh", this);
    //
    //this.onAfterRefresh.fire(this); //
    //
    //this.onAfterRefresh.subscribe(this.doParamChangeRefresh, this);
    //
    //this.onAfterRefresh.unsubscribe(this.doParamChangeRefresh, this);
}
lang.extend(SpreadsheetReport, "freequery.widget.Module2");

SpreadsheetReport.prototype.doParamChangeRefresh = function() {
    this.onAfterRefresh.unsubscribe(this.doParamChangeRefresh, this);
    this.doParamChangeNeedRefresh = false;
    var that = this;
    setTimeout(function() {
        //xx
    }, 1);
}
```

2.6Module2：js组件基类

完整名称：freequery.widget.Module2，可以是所有业务逻辑组件的基类，一般情况，编写界面是同名js文件和html文件配套，js文件是组件业务逻辑，一般会继承freequery.widget.Module2，html文件是布局文件，Module2内置了如下逻辑：

1、使用其中的addListener和removeListener方法给dom元素注册事件。

2、使用其中的init方法，实现了界面逻辑和界面布局的分离，界面布局可以是与组件js文件同目录及同名的.html，这样系统会自动加载布局文件，同时会给布局html文件中定义了bofid的元素自动执行以下操作：

（1）定义了bofid的dom元素，可以在js组件中通过**this.elem+bofid（其中首字母大写）**引用，譬如：，则this.elemTestSpan可以引用该元素

（2）js组件可以使用以下方法给元素添加事件：**elem + bofid（其中首字母大写）+ 事件名称+_handler**，如果在组件中添加命名符合这类规则的方法，系统会自动给对应元素加上对应鼠标事件，譬如elemTestSpan_click_handler，就是给bofid为testSpan的元素添加click事件，事件逻辑为命名符合规则的方法逻辑。

3、destroy方法，注销组件。

完整的示例见下面的2.7、BaseDialogEx：对话框组件基类。

方法参数及示例说明

```
/**
 * DOM,
 * this.addListener(this.elem_btnQuery , "click", this.refreshData , this);
 * @modifier final, protected
 * @param element
 *         DOM
 * @param type
 *         , 'click', 'mouseup'. : 'on'
 * @param handler
 *
 * @param that
 *         [] this
 * @param group
 *         [] . "", . removeListenersByGroup()
 * @return void
 */
Module2.prototype.addListener = function(element, type, handler, that, group) {}

/**
 * DOM,
 * this.removeListener(this.elem_btnQuery,"click",this.refreshData);
 * @modifier final, protected
 * @param element
 *         DOM
 * @param type
 *         , 'click', 'mouseup'. : 'on'
 * @param handler
 *
 * @param that
 *         [] this
 * @param group
 *         [] . addListener()
 * @return void
 */
Module2.prototype.removeListener = function(element, type, handler, that, group) {}

/**
 * @param container dom
 * @param url html__url jshtml
 * @param noWrapper true or falsefalse, containerhtmltrueHTMLcontainer
 * @param noDoGet html truehtmlfalsehtml
 */
Module2.prototype.init = function(container, url, noWrapper, noDoGet) {}
```

2.7BaseDialogEx: 对话框组件基类

freequery.dialog.BaseDialogEx继承了freequery.widget.Module2, 对话框内容组件基类, 主要方法:

1、init(parent, data, fn, obj, win): 初始化方法

参数说明:

- (1) **parent**: 窗口内容的父容器
- (2) **data**: 调用弹窗时传递给弹窗的数据
- (3) **fn**: 可选参数, 窗口关闭后回调函数
- (4) **obj**: 可选参数, 上面fn回调函数的this对象
- (5) **win**: 可选参数, 源窗口

2、destroy(): 注销方法

对话框内容组件需要配合dialogFactory.showDialog() 配合使用, 下面是其定义:

dialogFactory.showDialog定义

```
/**
 *
 *
 * @param dlgConf
 *
 * {fullName,size[option],width[option],height[option],resizable[option],title[option],
dialogType[option]}
 *
 *      fullName: ,,freequery.dialog.OpenSaveDialog <br>
 *      size: ,(width,height) <br>
 *      width: , <br>
 *      height: , <br>
 *      resizable: ,, 'yes', 'no' <br>
 *      title: , <br>
 *      dialogType: 'modal', 'modal', 'modeless' <br>
 * @param params
 *      initdata
 * @param fn
 *      ,initfn
 * @param obj
 *      this
 * @param opts
 *      eg: {success:function(){} }
 */
showDialog : function(dlgConf, params, fn, obj, opts) {}
```

下面是系统中点击[关于](#)的弹窗实现, AboutDialog.js是关于js组件, 其布局内容是AboutDialog.html, 二者通过AboutDialog.js: BaseDialogEx.superclass.init.call(this, this.dialogBody, __url, true)组合在一起。

AboutDialog.js

```
var BaseDialogEx = jsloader.resolve("freequery.dialog.BaseDialogEx");
var domutils = jsloader.resolve("freequery.lang.domutils");
var util = jsloader.resolve("freequery.common.util");
var PagePanel = jsloader.resolve("freequery.control.PagePanel");
var Configuration = jsloader.resolve("Configuration");

var AboutDialog = function() {
};
lang.extend(AboutDialog, BaseDialogEx);

AboutDialog.prototype.init = function(parent, data, fn, obj) {
    AboutDialog.superclass.init.call(this, parent, data, fn, obj);
    BaseDialogEx.superclass.init.call(this, this.dialogBody, __url, true);
    this.dialogBody.style.border = 'none';
    this.dialogBody.style.backgroundColor = 'transparent';
    document.body.style.overflow = "";
    var companyName = data[0];
    var webAddress = data[1];
    var mailAddr = data[2];
    this.parentWindow = data[3]
    this.setButtonVisible("BTNCANCEL", false);
    this.lblCompanyName = domutils.findElementByClassName(this.parent, "_companyname");
    if (companyName)
        this.lblCompanyName.innerHTML = companyName;
    this.lblWebAddress = domutils.findElementByClassName(this.parent, "_webaddress");
    if (webAddress) {
        this.lblWebAddress.innerHTML = webAddress;
        this.lblWebAddress.href = webAddress;
    }
    if (mailAddr) {
        this.elemMailto.href = mailAddr;
    }
    this.buildDate = domutils.findElementByClassName(this.parent, "_buildDate");
    if (Configuration.isSpreadsheetEdition || Configuration.isXQueryEdition) {
        var trBuild = this.buildDate.parentNode.parentNode;
    }
}
```

```

        var tbody = trBuild.parentNode;
        var trEdition = tbody.insertRow(trBuild.rowIndex);
        var tdEdition = trEdition.insertCell(-1);
        tdEdition.style.height = '20px';
        tdEdition.style.textAlign = 'center';
        var name = Configuration.isSpreadsheetEdition ? 'Spreadsheet' : 'xQuery';
        var text = 'Smartbi ' + name + ' Edition';
        tdEdition.innerText = text;
    }
    this.warBuildDate = domutils.doGet("version.txt");
    if (this.warBuildDate.indexOf("HTTP Status 404") >= 0) {
        this.warBuildDate = "${Mayistheddevelopmentversion}";
    }
    this.warPackageInfo = domutils.doGet("packageinfo.txt");
    this.warSourceVersion = "";
    this.warSourceTag = "";
    if (this.warPackageInfo.indexOf("HTTP Status 404") == -1) {
        var versionInfo = /\b(Version:.+)(\n|\r)/.exec(this.warPackageInfo);
        if (versionInfo != null) {
            this.warSourceVersion = versionInfo[0];
        }
        var tagInfo = /\b(TAG:.+)(\n|\r)/.exec(this.warPackageInfo);
        if (tagInfo != null) {
            this.warSourceTag = tagInfo[0];
        }
    }
    this.buildDate.innerHTML = this.warBuildDate + "<br />" + this.warSourceVersion + "<br />"
        + this.warSourceTag;
    this.initTab();
}

AboutDialog.prototype.destroy = function() {
    if (this.aboutTab) {
        this.aboutTab.destroy();
    }
    if (this.licenseInfoTab) {
        this.licenseInfoTab.destroy();
    }
    if (this.javaInfoTab) {
        this.javaInfoTab.destroy();
    }
    if (this.pagecontrol) {
        this.pagecontrol.destroy();
    }
    AboutDialog.superclass.destroy.call(this);
}

AboutDialog.prototype.initTab = function() {
    this.initLicenseInfo(this.elemLicenseInfoDiv);
    this.initJavaInfo(this.elemJavaInfoDiv);
    this.pagecontrol = new PagePanel(this.elemTabPanel);
    this.pagecontrol.contentBanner.style.backgroundColor = '#fff';
    this.aboutTab = this.pagecontrol.appendTab();
    this.aboutTab.setCaption("${About}");
    this.licenseInfoTab = this.pagecontrol.appendTab();
    this.licenseInfoTab.setCaption("License");
    this.javaInfoTab = this.pagecontrol.appendTab();
    this.javaInfoTab.setCaption("Java");
    this.aboutTab.appendItem(this.elemAboutDiv);
    if (this.aboutTab.itemParent) {
        this.aboutTab.itemParent.style.verticalAlign = 'middle';
    }
    this.licenseInfoTab.appendItem(this.elemLicenseInfoDiv);
    this.javaInfoTab.appendItem(this.elemJavaInfoDiv);
    this.pagecontrol.tabs[0].setActive();
    this.pagecontrol.autoFitResize();
}

AboutDialog.prototype.initLicenseInfo = function(elem) {
    var func = function(ret) {
        if (ret && ret.succeeded && ret.result) {

```



```

        var result = ret.result;
        if (result.hasLicense) {
            var html = [ '<table style="table-layout: fixed;" cellPadding="0"
cellSpacing="0">' ];

            html.push('<col style="width:45%" /><col style="width:55%" />');
            html.push('<tr><td>${License.Type}${Colon}</td><td>'
                + (result.type == 'evaluation' ? "${License.Evaluation}"
                    : "${License.Formal}") + "</td></tr>");
            html.push('<tr><td>${License.Licensee}${Colon}</td><td>' + result.licensee +
"</td></tr>");
            html.push('<tr><td>${License.Expiration}${Colon}</td><td>' + result.
                expiration
                + "</td></tr>");
            html.push('<tr><td>${License.AuthorizedUsersCount}${Colon}</td><td>'
                + this.parseCount(result.authorizedUsers) + "</td></tr>");
            html.push('<tr><td>${License.SessionCount}${Colon}</td><td>'
                + this.parseCount(result.sessionCount) + "</td></tr>");
            html.push('<tr><td>${License.ReportCount}${Colon}</td><td>'
                + this.parseCount(result.reportCount) + "</td></tr>");
            html.push('<tr><td>${License.MobileCount}${Colon}</td><td>'
                + this.parseCount(result.mobile) + "</td></tr>");
            html.push("</table>");
            elem.innerHTML = html.join('');
            this.elemLicenSeto.innerHTML = "${LicenSeto}" + result.licensee;
        } else {
            elem.innerHTML("${License.NoLicense}");
        }
    }
};
util.remoteInvoke("CommonService", "getLicenseInfo", [], func, this);
}

AboutDialog.prototype.parseCount = function(count) {
    return (count == -1) ? "${License.Unlimited}" : count;
}

AboutDialog.prototype.initJavaInfo = function(elem) {
    var func = function(ret) {
        if (ret && ret.succeeded && ret.result) {
            var result = ret.result;
            var html = [];
            html.push('<div class="wrapper-outer"><div class="wrapper-inner">');
            html.push('<table style="empty-cells: show; width:100%; height:100%; table-layout:
fixed;" cellPadding="0" cellSpacing="0">');
            html.push('<col style="width:35%" /><col />');
            var tdStyle = ' style="word-break: break-all; word-wrap:break-word;" ';
            for (var i = 0, len = result.length; i < len; i++) {
                var row = result[i];
                html.push('<tr><td ' + tdStyle + '>' + row[0] + '</td><td ' + tdStyle + '>'
                    + row[1] + '</td></tr>');
            }
            html.push('</table>');
            html.push('</div></div>');
            elem.innerHTML = html.join('');
        }
    };
    util.remoteInvoke("CommonService", "getSystemProperties", [], func);
}

AboutDialog.prototype.elemProductWeb_click_handler = function(e) {
    domutils.stopEvent(e);
    this.parentWindow.open(this.lblWebAddress.href || this.lblWebAddress.getAttribute("href"));
};

AboutDialog.prototype.elemReleaseNotes_click_handler = function(e) {
    domutils.stopEvent(e);
    this.parentWindow.open(this.elemReleaseNotes.href || this.elemReleaseNotes.getAttribute("href"));
};

AboutDialog.prototype.elemSupportWeb_click_handler = function() {
};

```

```
AboutDialog.prototype.elemCompareName_click_handler = function(e) {
    domutils.stopEvent(e);
    this.parentWindow.open(this.lblWebAddress.href || this.lblWebAddress.getAttribute("href"));
};
```

AboutDialog.html

```
<div align="center" style="width:100%; height:100%; line-height:22px;">
    <div bofid="tabPanel" class="_tabPanel" style="height:100%;width:100%;"></div>
    <table bofid="aboutDiv" class="_list_log_div tab-content system_color_white " style="overflow:auto;"
width="100%" border="0">
        <tr>
            <td width="100%" valign="top">
                <table width="100%" border="0" cellspacing="0" cellpadding="0" align="center">
                    <tr>
                        <td height="73px" align="center" style="background:url(img/about/logo${ImageSuffix}.png)
no-repeat center center;"></td>
                    </tr>
                    <tr>
                        <td height="20" align="center">
                            <span bofid="licenseto"></span>
                        </td>
                    </tr>
                    <tr>
                        <td height="20" align="center">
                            <span>Build:&nbsp;</span><span class="_buildDate">2008-10-31</span>
                        </td>
                    </tr>
                    <tr>
                        <td height="20" align="center">
                            <span class="_companyname">${Smartbi}</span>&nbsp;</td>
                    </tr>
                    <tr>
                        <td height="20" align="center">
                            <a class="_webaddress link-item" bofid="compareName" target="_blank" href="
http://www.smartbi.com.cn">http://www.smartbi.com.cn</a>
                        </td>
                    </tr>
                    <tr>
                        <td height="20" align="center">
                            <a class="link-item2" bofid="productWeb" target="_blank" href="http://www.smartbi.com.cn"
>${ProductWebsite}</a>&nbsp;</td>
                    </tr>
                    <tr>
                        <td height="20" align="center">
                            <a class="link-item2" bofid="releaseNotes" target="_blank" href="https://history.wiki.
smartbi.com.cn/pages/viewpage.action?pageId=17956904">${ReleaseNotes}</a>&nbsp;</td>
                    </tr>
                    <tr>
                        <td height="20" align="center">
                            <a class="link-item2" bofid="mailto" href="mailto:support@smartbi.com.cn"
>${Contactus}</a>
                        </td>
                    </tr>
                </table>
            </td>
        </tr>
    </table>
    <table width="100%" border="0" >
        <tr>
            <td>
                <table>
                    <tr>
                        <td>
                            <div bofid="licenseInfoDiv" class="editblock _list_run_div tab-content system_color_white " style="
overflow:auto; height:100%;width:100% "/>
                            <div bofid="javaInfoDiv" class="editblock _list_run_div tab-content system_color_white " style="
overflow:auto; height:100%; width:100% "/>
                        </td>
                    </tr>
                </table>
            </td>
        </tr>
    </table>
```

显示对话框方法:

显示AboutDialog

```
BannerView.prototype.showAbout = function() {  
    var data = [ registry.get('CompanyName'), registry.get('WebAddress'), registry.get('MailAddr'),  
window ];  
    var dialogConfig = {  
        title : '${About} Smartbi',  
        size : DialogFactory.getInstance().size.MIDDLE,  
        fullName : 'freequery.main.AboutDialog'  
    };  
    DialogFactory.getInstance().showDialog(dialogConfig, data);  
};
```