

自定义对话框

- 1 概述
- 2 自定义对话框的基本对象
 - BaseDialogEx
 - dialogFactory
- 3 创建自定义对话框的步骤
 - 步骤一：创建对话框业务处理类
 - 步骤二：实现init和destroy方法
 - 步骤三：调用父类的init和destroy方法
 - 步骤四：在init方法中执行自定义逻辑
 - 步骤五：显示指定的对话框
- 4 常用开发技巧
 - 4.1 使用HTML模板设计对话框内容
 - 4.2 对话框关闭时执行回调函数
- 5 视频教学

1 概述

在Smartbi中，有较多场景会用到对话框。本文介绍在Smartbi中创建自定义对话框的方法。

目前Smartbi的对话框底层采用layer弹层组件实现。经过Smartbi对layer的封装，使得对话框的应用更加便捷，同时也更方便根据需要替换底层的弹层组件。如下图所示：



2 自定义对话框的基本对象

创建自定义对话框前，需要了解对话框组件的基类：**BaseDialogEx**，以及对话框的工具类：**dialogFactory**。

BaseDialogEx

对话框组件基类freequery.dialog.BaseDialogEx继承了freequery.widget.Module2，主要方法包括两个：初始化方法（init）、注销方法（destroy）

1) 初始化方法：**init**

- 定义：BaseDialogEx.prototype.init = function(parent, data, fn, obj, win)
- 功能：初始化一个空的对话框。
- 参数：@parent（对话框内容的父容器）；@data（调用对话框时传递给对话框的数据）；@fn（可选参数，对话框关闭后回调函数）；@obj（可选参数，上面fn回调函数的this对象） @win（可选参数，源窗口）。

2) 注销方法：**destroy**

- 定义：BaseDialogEx.prototype.destroy = function()
- 功能：关闭对话框，销毁相关资源。

dialogFactory

对话框工具类：freequery.dialog.dialogFactory，主要包含showDialog方法。

1) 显示对话框方法：**showDialog**

- 定义：showDialog : function(dlgConf, params, fn, obj, opts)
- 功能：根据传入的对话框配置，弹出一个对话框。
- 参数：@dlgConf（对话框配置对象）；@params（需要传递到对话框的数据）；@fn（对话框关闭的回调函数）；@obj（回调函数内部的this对象）；@opts（扩展配置，如：{ success: function() {} }）。

2) 详细参数说明：@dlgConf：对话框配置对象，对象格式{fullName:'xx.xx.xx',size:[100,200],title:'***' }

常用配置项如下：

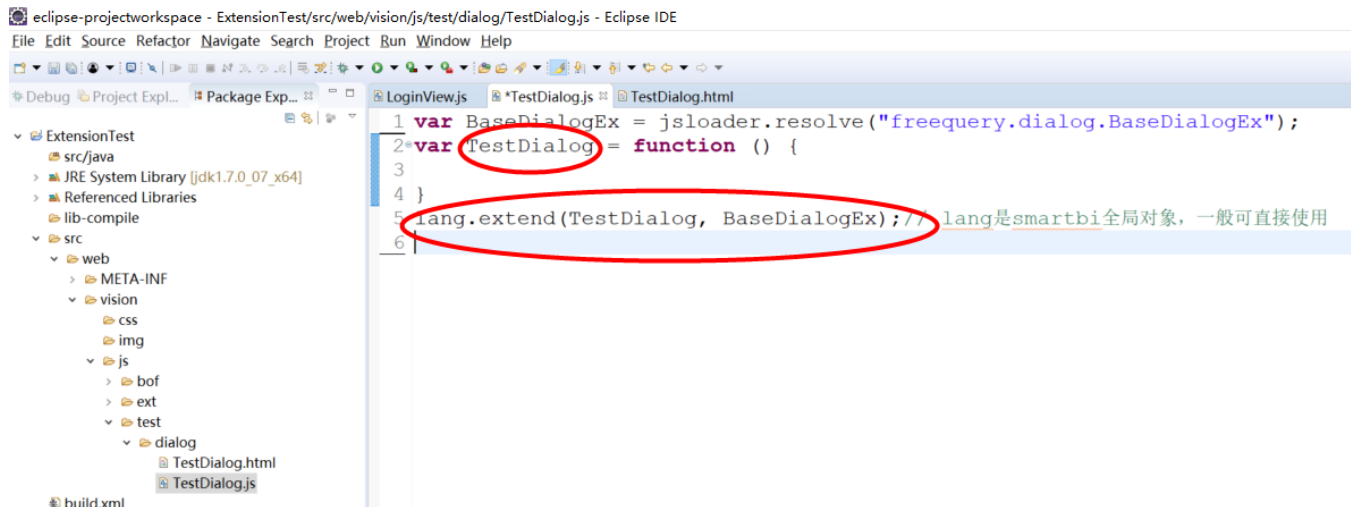
- fullName：必选项，对话框的全名，例如 freequery.dialog.OpenSaveDialog
- size：可选项，对话框尺寸（指定该项将会忽略width和height）
- width：可选项，对话框宽度
- height：可选项，对话框高度
- resizable：可选项，对话框大小是否可变，合法值 'yes'， 'no'
- title：可选项，对话框标题
- dialogType：可选项，对话框模态，缺省值'modal'，合法值 'modal'， 'modeless'

3 创建自定义对话框的步骤

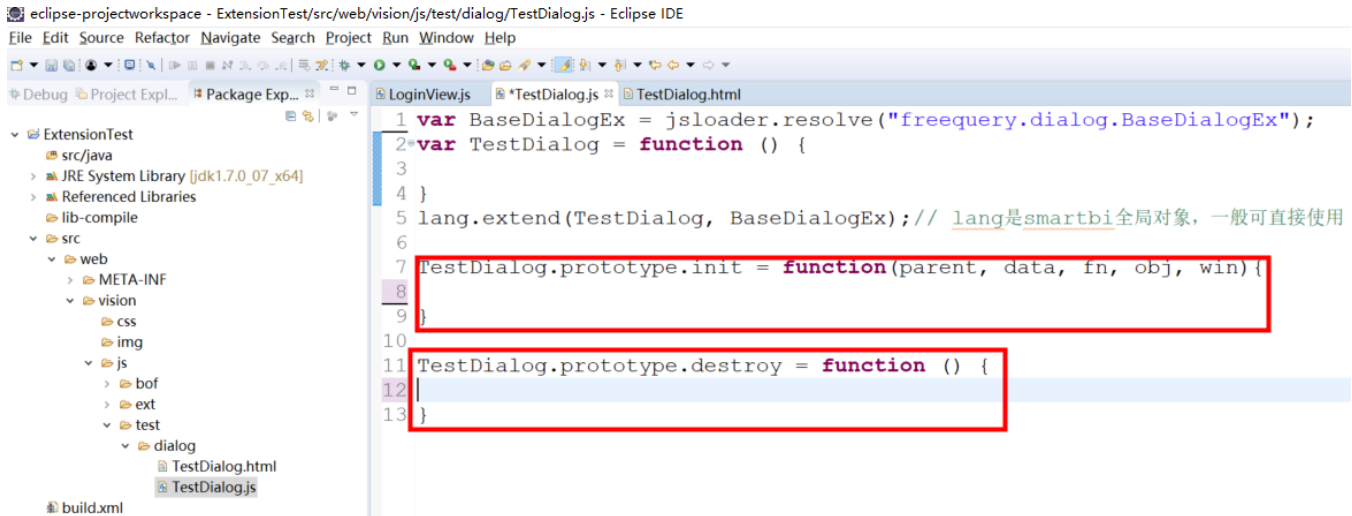
创建自定义对话框的**整体步骤**：创建一个普通的JS类，使其继承BaseDialogEx；实现init和destroy方法，并分别调用父类BaseDialogEx的init和destroy方法；最后采用dialogFactory工具类提供的showDialog方法显示对话框即可。 详细操作步骤如下：

步骤一：创建对话框业务处理类

创建一个对话框的业务处理JS类，使其继承BaseDialogEx类。 如：创建test.dialog.TestDialog.js类：



步骤二：实现init和destroy方法



步骤三：调用父类的init和destroy方法

- 1) 在init初始化方法中调用父类BaseDialogEx的init初始化方法，用于创建一个空的对话框。

```
TestDialog.prototype.init = function(parent, data, fn, obj, win){
    // 调用父类BaseDialogEx的init方法
    TestDialog.superclass.init.call(this, parent, data, fn, obj, win);
}
```

- 2) 在destroy方法中调用父类的BaseDialogEx的destroy注销方法，用于关闭对话框及其他相关资源。

```
TestDialog.prototype.destroy = function () {
    TestDialog.superclass.destroy.call(this); // 调用父类BaseDialogEx的destroy方法
}
```

步骤四：在init方法中执行自定义逻辑

在执行完父类BaseDialogEx的init初始化方法后，会获取到一个空的对话框，其实还会创建几个可直接使用的全局对象，包括：this.dialogBody（当前对话框内容显示区域的根节点）、this.btnOK（确定按钮）、this.btnCancel（取消按钮）等。

如果想自定义显示内容，则可在init方法中编写自定义逻辑，如，最简单的，直接显示data传递的内容。这样就创建完成一个自定义对话框了。

```
TestDialog.prototype.init = function(parent, data, fn, obj, win){
    debugger;
    // 调用父类BaseDialogEx的init方法
    TestDialog.superclass.init.call(this, parent, data, fn, obj, win);
    this.dialogBody.innerHTML = data[0];
}
```

步骤五：显示指定的对话框

通过Smartbi封装的对话框工具类dialogFactory.js可以显示指定的对话框。以在报表宏中显示对话框为例（在JS文件中调用方式相同），示例如下：

名称: 对话框 类型: ClientSide 对象: simpleReport 事件: onRenderTable

```
1 function main(simpleReport, simpleReportContext) {
2     var grid = simpleReport.grid;
3     var cell = grid.getCell(1, 1);
4     cell.innerHTML = "";
5     cell.innerHTML = '<button class = "_btnTest barbtn btn-default" style = "background-color: #00ccff; color: white;" >测试</button>';
6     var dialogBtn = domutils.getElementsByClassName("_btnTest barbtn btn-default", cell)[0];
7     cell.appendChild(dialogBtn);
8     simpleReport.addListener(dialogBtn, "click", dialogTest, simpleReport);
9 }
10
11 function dialogTest() {
12     var DialogFactory = imports('freequery.dialog.dialogFactory');
13     var data = ["HelloWorld!"];
14     var dialogConfig = {
15         title: '测试自定义对话框',
16         size: DialogFactory.getInstance().size.MIDDLE,
17         fullName: 'test.dialog.TestDialog'
18     };
19     DialogFactory.getInstance().showDialog(dialogConfig, data, simpleReport);
20 }
```

☐ 禁用

格式化代码(F)

保存(S)

关闭(C)

最终效果如下:



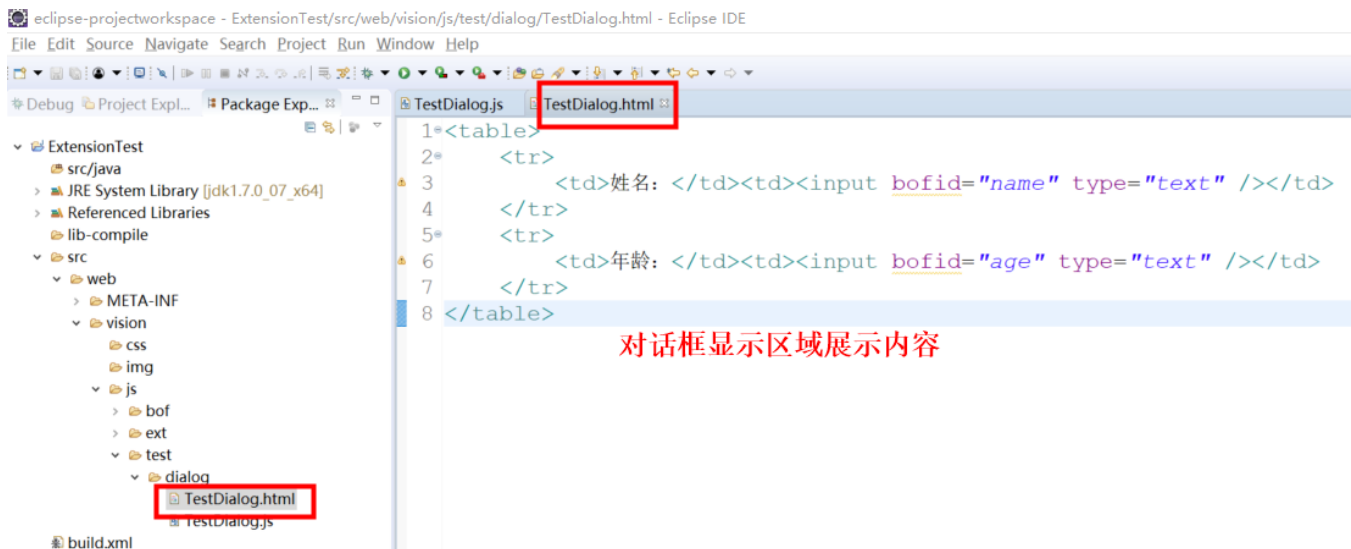
4 常用开发技巧

4.1 使用HTML模板设计对话框内容

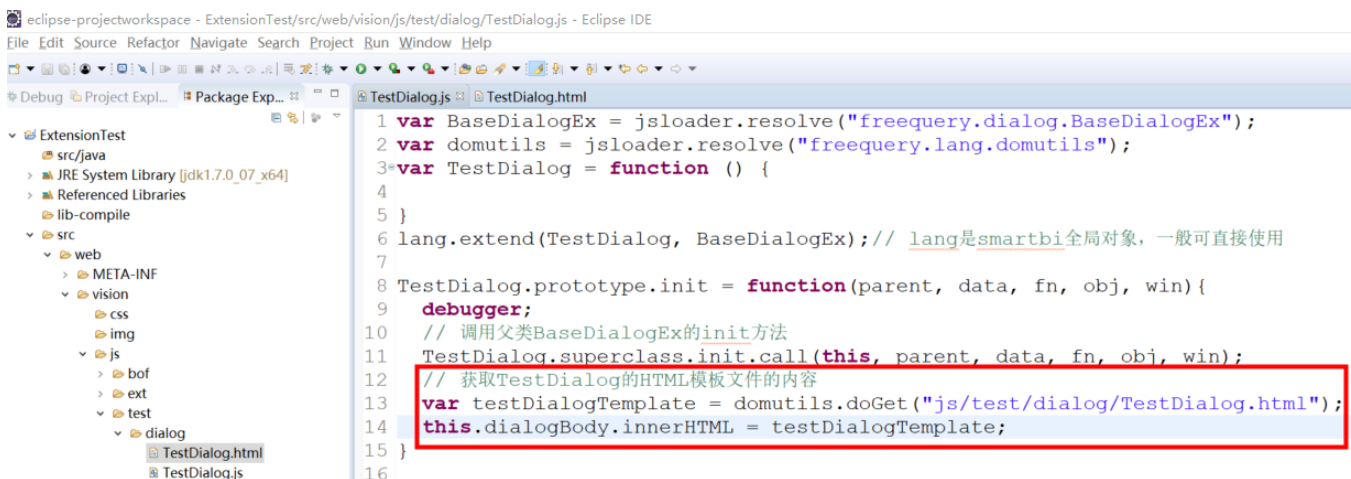
在实际应用对话框时，经常会需要开发比较复杂的对话框展示内容。如果全部使用JS代码在init方法中动态创建DOM元素去展示的话将非常麻烦。

此时我们可以结合Smartbi中JS组件类与HTML布局分离的机制，使用HTML模板文件，编写对话框展示的内容，然后采用JS实现业务逻辑的控制。使用示例如下：

- 1) 编写HTML模板



2) init方法中获取HTML模板内容，添加到this.dialogBody中



注意：获取模板时，也可采用Module2.js中的init方法，自动加载并解析HTML模板内容，后续可直接根据模板中的bofid值获取到指定元素（规则：this.elem+bofid值，bofid值首字母大写） 调用语句：BaseDialogEx.superclass.init.call(this, this.dialogBody, __url, true);

```

*TestDialog.js  TestDialog.html
1 var BaseDialogEx = jsloader.resolve("freequery.dialog.BaseDialogEx");
2 var domutils = jsloader.resolve("freequery.lang.domutils");
3 var TestDialog = function () {
4
5 }
6 lang.extend(TestDialog, BaseDialogEx); // lang是smartbi全局对象，一般可直接使用
7
8 TestDialog.prototype.init = function(parent, data, fn, obj, win){
9     debugger;
10    // 调用父类BaseDialogEx的init方法
11    TestDialog.superclass.init.call(this, parent, data, fn, obj, win);
12    // 获取TestDialog的HTML模板文件的内容
13    // var testDialogTemplate = domutils.doGet("js/test/dialog/TestDialog.html");
14    // this.dialogBody.innerHTML = testDialogTemplate;
15    BaseDialogEx.superclass.init.call(this, this.dialogBody, __url, true);
16 }
17
18 TestDialog.prototype.destroy = function () {
19     TestDialog.superclass.destroy.call(this); // 调用父类BaseDialogEx的destroy方法
20 }

```

3) 对话框中即可展示HTML模板文件中定义的内容。

测试自定义对话框

×

姓名:

年龄:

确定(O)

取消(C)

4.2 对话框关闭时执行回调函数

在调用dialogFactory.showDialog方法时，可传递回调函数，在对话框关闭时执行，实现对话框关闭后执行相关操作。

使用示例如下：传入callback函数作为对话框关闭后的回调函数。回调函数内判断用户是否点击了确定按钮，是则刷新报表，否则不做处理。

名称: 自定义对话框 类型: ClientSide 对象: simpleReport 事件: onRenderTable

```
1 function main(simpleReport, simpleReportContext) {
2   var grid = simpleReport.grid;
3   var cell = grid.getCell(1, 1);
4   cell.innerText = "";
5   cell.innerHTML = '<button class = "_btnTest barbtn btn-default" style = "background-color: #00ccff; color: white;" >测试</button>';
6   var dialogBtn = domutils.getElementsByClassName("_btnTest barbtn btn-default", cell)[0];
7   cell.appendChild(dialogBtn);
8   simpleReport.addListener(dialogBtn, "click", dialogTest, simpleReport);
9 }
10
11 function dialogTest() {
12   var DialogFactory = imports('freequery.dialog.dialogFactory');
13   var data = [];
14   var dialogConfig = {
15     title: '测试自定义对话框',
16     size: DialogFactory.getInstance().size.MIDDLE,
17     fullName: 'test.dialog.TestDialog'
18   };
19   DialogFactory.getInstance().showDialog(dialogConfig, data, callback, simpleReport);
20 }
21
22 function callback(ret) {
23   if (ret === true) {
24     this.refreshData();
25   }
26 }
```

5 视频教学

视频教学点击下载：[自定义对话框\(part1\)](#)、[自定义对话框\(part2\)](#)