

第二课：客户端宏

- 1. 概述
- 2. 客户端宏原理
- 3. 客户端宏事件
 - spreadsheetReport（电子表格对象）
 - simpleReport（即席查询/灵活分析对象）
 - INSIGHT（透视分析对象）
 - meterReport（仪表分析对象）
 - olapQuery（多维分析报表对象）
 - olapTable（多维分析表格对象）
- 4. 客户端宏API和源码
- 5. 客户端宏示例
- 6. 视频教学

1. 概述

Smartbi提供客户端宏，用以实现特定的报表展现效果，因为仅在页面展示时生效，所以称为客户端宏（如下图，页面渲染时在参数后面添加特定按钮）。



在进行客户端宏的学习前，建议了解：

- [报表宏创建方式](#)：在Smartbi中如何创建客户端宏，并创建一个简单的宏示例。
- [跳转规则](#)：根据跳转规则生成的宏脚本本具有很大的参考意义，节省编写宏的时间。

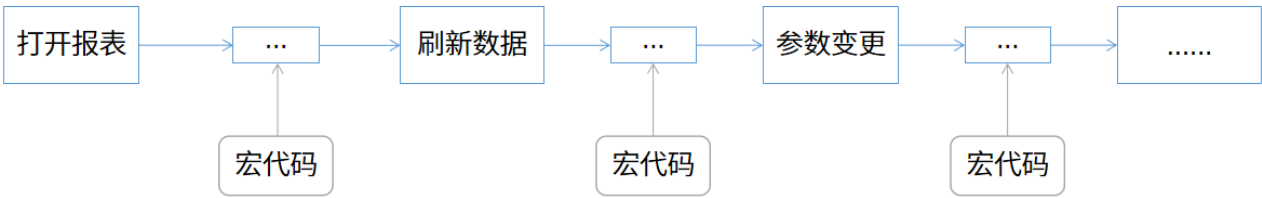
注意：

- 1) 客户端宏主要影响浏览器端的展现效果，除了图形（直接加在图形对象上的部分宏功能可以对导出生效），其他报表的客户端宏都不会对导出有效果。
- 2) 客户端宏是在浏览器端运行，所以语法是JavaScript的语法，需要有前端开发的经验才可能灵活编写，调试也是使用浏览器前端调试即可。

2. 客户端宏原理

Smartbi的报表从打开到完整展现的过程，会经历一系列的步骤（事件），比如打开报表，刷新报表，参数改变等。在进行这些步骤的前后，通过加入扩展的入口，使得程序在执行这些步骤前后可以加载对应事件的自定义宏代码，以达到在对应事件中方便、准确地修改报表的目的。

如下图所示，报表打开后会触发宏代码，刷新数据后会触发宏代码，参数变更后同样会触发宏代码。



3. 客户端宏事件

ClientSide客户端模块

spreadsheetReport （电子表格对象）

spreadsheetReport：电子表格对象。

对应事件	对应事件说明
onRenderReport	<ul style="list-style-type: none">• 事件原型：function main(spreadsheetReport) {}• 事件说明：在打开电子表格报表初始化完成后触发• 参数说明<ul style="list-style-type: none">o spreadsheetReport：电子表格报表对象• 参考示例：电子表格报表定时刷新
onRender	<ul style="list-style-type: none">• 事件原型：function main(spreadsheetReport) {}• 事件说明：在电子表格报表数据刷新后触发• 参数说明<ul style="list-style-type: none">o spreadsheetReport：电子表格报表对象• 参考示例：电子表格保存按钮位置修改
onParamValueChanged	<ul style="list-style-type: none">• 事件原型：function main(spreadsheetReport, param) {}• 事件说明：在电子表格报表参数改变后触发• 参数说明<ul style="list-style-type: none">o spreadsheetReport：电子表格报表对象o param当前改变的参数对象• 参考示例：电子表格根据参数值不同显示不同的报表
onLinkClick	<ul style="list-style-type: none">• 事件原型：function main(spreadsheetReport, ruleName, params) {}• 事件说明：在电子表格超链接上点击后触发，这个事件需要配合服务端宏才能生效，一般不会手动新建，而是通过跳转规则自动生成（可以基于生成的再修改）。• 参数说明<ul style="list-style-type: none">o spreadsheetReport：电子表格报表对象o ruleName：跳转规则名称o params：报表参数• 参考示例：电子表格不同角色跳转到不同的目标报表

onChartClick	<ul style="list-style-type: none">• 事件原型: function main(spreadsheetReport, spreadsheetReportChart) {}• 事件说明: 在电子表格图形上点击后触发，这个事件需要配合服务端宏才能生效，一般不会手动新建，而是通过跳转规则自动生成（可以基于生成的再修改）。• 参数说明<ul style="list-style-type: none">o spreadsheetReport: 电子表格报表对象o spreadsheetReportChart: 被点击图形名称（excel设计器上指定的），对象结构，spreadsheetReportChart.name代表图形名称
onWriteBackInsertedRow	<ul style="list-style-type: none">• 事件原型: function main(spreadsheetReport, editingCellPosition) {}• 事件说明: 在可回写的电子表格的工具类上点击“插入行”后触发• 参数说明<ul style="list-style-type: none">o spreadsheetReport: 电子表格报表对象o editingCellPosition: 当前编辑单元格，对象结构{row:, column:}• 参考示例: 电子表格添加新行时自动为单元格赋值
onWriteBackCellValueChanged	<ul style="list-style-type: none">• 事件原型: function main(spreadsheetReport, editingCellPosition, oldValue, newValue, newDisplayValue) {}• 事件说明: 在可回写的电子表格中，修改某个单元格值的时候触发• 参数说明<ul style="list-style-type: none">o spreadsheetReport: 电子表格报表对象o editingCellPosition: 当前编辑单元格，对象结构{row:, column:}o oldValue: 修改前的值o newValue: 修改后的值o newDisplayValue: 修改后的显示值• 参考示例: 电子表格回写单元格一对多联动
onWriteBackSavedData	<ul style="list-style-type: none">• 事件原型: function main(spreadsheetReport, succeeded) {}• 事件说明: 在可回写的电子表格的工具类上点击“保存”，服务器返回后触发• 参数说明<ul style="list-style-type: none">o spreadsheetReport: 电子表格报表对象o succeeded: bool，回写是否成功• 参考示例: 电子表格回写保存时记录日志

simpleReport（即席查询/灵活分析对象）

simpleReport: 即席查询/灵活分析对象。

对应事件	对应事件说明
onRender	<ul style="list-style-type: none">• 事件原型: function main(simpleReport, simpleReportContext) {}• 事件说明: 在打开报表，初始化完成后触发• 参数说明<ul style="list-style-type: none">o simpleReport: 即席查询/灵活分析对象o simpleReportContext: 即席查询/灵活分析上下文对象• 参考示例: 即席查询如何实现隐藏工具栏按钮

onRenderTable	<ul style="list-style-type: none">• 事件原型: <code>function main(simpleReport, simpleReportContext) {}</code>• 事件说明: 报表刷新数据后触发，每次刷新数据都触发一次• 参数说明<ul style="list-style-type: none">o <code>simpleReport</code>: 即席查询/灵活分析对象o <code>simpleReportContext</code>: 即席查询/灵活分析上下文对象• 参考示例: 即席查询如何实现动态标题
onParamValueChanged	<ul style="list-style-type: none">• 事件原型: <code>function main(simpleReport, simpleReportContext, param) {}</code>• 事件说明: 报表参数改变后触发• 参数说明<ul style="list-style-type: none">o <code>simpleReport</code>: 即席查询/灵活分析对象o <code>simpleReportContext</code>: 即席查询/灵活分析上下文对象o <code>param</code>: 发生变化的参数对象• 参考示例: 即席查询如何实现根据不同参数值显示不同的报表

INSIGHT（透视分析对象）

INSIGHT：透视分析对象。

对应事件	对应事件说明
onRender	<ul style="list-style-type: none">• 事件原型: <code>function main(insight) {}</code>• 事件说明: 在打开透视分析初始化完成后触发• 参数说明<ul style="list-style-type: none">o <code>insight</code>: 透视分析对象• 参考示例: 透视分析表格如何实现屏蔽右键菜单
onRenderTable	<ul style="list-style-type: none">• 事件原型: <code>function main(insight) {}</code>• 事件说明: 透视分析刷新数据后触发，每次刷新数据都触发一次• 参数说明<ul style="list-style-type: none">o <code>insight</code>: 透视分析对象• 参考示例: 透视分析打开默认收缩
onParamValueChanged	<ul style="list-style-type: none">• 事件原型: <code>function main(insight, param) {}</code>• 事件说明: 透视分析参数改变后触发• 参数说明<ul style="list-style-type: none">o <code>insight</code>: 透视分析对象o <code>param</code>: 发生变化的参数对象

meterReport（仪表分析对象）

meterReport：仪表分析对象。

对应事件	对应事件说明
onRender	<ul style="list-style-type: none">• 事件原型: <code>function main(meterReport) {}</code>• 事件说明: 在打开仪表分析初始化完成后，刷新数据前触发• 参数说明<ul style="list-style-type: none">o <code>meterReport</code>: 仪表分析对象

onParamValueChanged	<ul style="list-style-type: none">• 事件原型: function main(meterReport, param) {}• 事件说明: 仪表分析参数改变后触发• 参数说明<ul style="list-style-type: none">o meterReport: 仪表分析对象
---------------------	---

olapQuery（多维分析报表对象）

olapQuery：多维分析报表对象。

对应事件	对应事件说明
afterRefreshTable	<ul style="list-style-type: none">• 事件原型: function main(olapQuery) {}• 事件说明: 在多维分析报表数据刷新后触发• 参数说明<ul style="list-style-type: none">o olapQuery: 多维分析报表对象• 参考示例: 多维分析如何实现数据标注效果

olapTable（多维分析表格对象）

olapTable：多维分析报表表格对象。

对应事件	对应事件说明
onClick	<ul style="list-style-type: none">• 事件原型: function main(cell) {}• 事件说明: 在单击多维分析表格中的内容时触发• 参数说明<ul style="list-style-type: none">o cell: 当前点击的单元格对象
onInitPopupMenu	<ul style="list-style-type: none">• 事件原型: function main(popupMenu, cell) {}• 事件说明: 在单击多维分析表格进行该事件处理，显示命令菜单，只有单击菜单中相关命令才会执行宏• 参数说明<ul style="list-style-type: none">o popupMenu: 子菜单对象o cell: 当前点击的单元格对象

4. 客户端宏API和源码

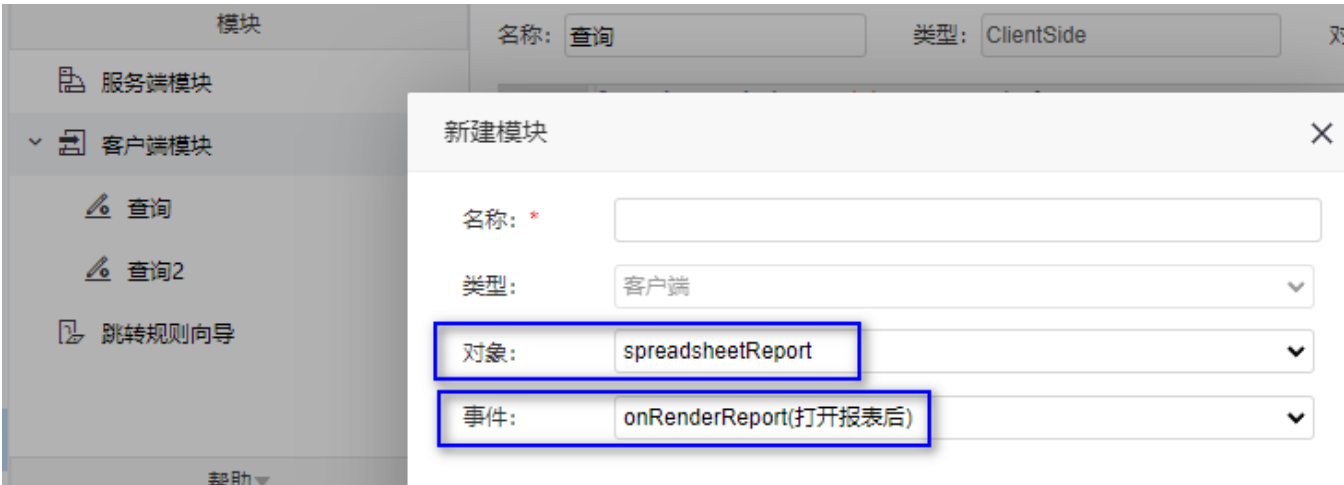
宏的开发实际离不开API（[报表宏接口文档](#)），根据API可以用报表宏快速实现多种功能。而要更加灵活开发则需要了解每个宏对象对应的源码，宏对象对应JS源文件如下（文件位于Smartbi war包中）：

报表类型	宏对象	源文件
电子表格	spreadsheetReport	SpreadsheetReport.js
即席查询	simpleReport	CombinedQV.js（继承QueryView.js）
	simpleReportContext	CombinedQVNavigator.js（继承QueryNavigator.js）
灵活分析	simpleReport	QueryView.js
	simpleReportContext	QueryNavigator.js
透视分析	INSIGHT	Insight.js
仪表分析	meterReport	Dashboard.js
多维分析	olapQuery	OlapQueryViewDisplay.js 或 OlapQueryView.js
图形	chartView	EChartsView.js 以及父类 AbstractChartView.js

5. 客户端宏示例

示例需求：在电子表格的参数后面添加查询按钮，点击按钮实现报表刷新。

需求是在参数后面添加按钮，即添加页面元素，所以应该使用客户端宏。因为是只添加一次的按钮，所以选择onRenderReport事件，在报表初始化完成后使用宏添加指定按钮，并给按钮添加事件，实现报表刷新。具体代码如下：



电子表格参数面板添加按钮

```
function main(spreadsheetReport) {
    addButton(spreadsheetReport);
}

//
function addButton(spreadsheetReport) {
    // ,
    var paramTable = spreadsheetReport.paramPanelObj.layoutTable;
    var index = paramTable.rows.length > 0 ? paramTable.rows.length - 1 : 0;
    var lastTr = paramTable.rows[index];
    //
    var btnPanel = lastTr.insertCell(-1);

    if (!spreadsheetReport.refreshBtn) {
        //
        var refreshBtn = createBtn("", spreadsheetReport.doRefresh, spreadsheetReport);
        btnPanel.appendChild(refreshBtn);
        spreadsheetReport.refreshBtn = refreshBtn;
    }
}

//
function createBtn(text, handler, spreadsheetReport) {
    var btn = document.createElement("button");
    btn.type = "button";
    btn.innerText = text;
    btn.title = text;
    btn.className = "barbtn";
    btn.style.width = "65px";
    //
    spreadsheetReport.addListener(btn, "click", handler, spreadsheetReport);
    return btn;
}
```

执行效果如下图所示：

III

🏠 导航 | 电子表格示例 x

🔄

☆

📄 导出

🖨️

👤 个人参数

开始日期*

2017-01-01

▼

结束日期*

2017-01-01

▼

查询

所属省	所属市	日期	数量
湖北	武汉市	2017-01-01	210

更多客户端宏示例详见：[客户端宏示例](#)。

6. 视频教学

视频教学点击下载：[客户端宏原理](#)、[客户端宏事件](#)、[客户端宏对象](#)、[客户端宏实战](#)

视频配套资源下载：[客户端宏示例](#)