


电子表格跳转既传递实际值又显示显示值

 提示：本文档的示例代码仅适用于本文档中的示例报表/场景。若实际报表/场景与示例代码无法完全适配（如使用功能不一致，或多个宏代码冲突等），需根据实际需求开发代码。

应用场景： 在项目中跳转传参，一般都是传编码、ID等值，但是显示给客户的所见的又要是有意义的、最终用户能懂得的名称，所以在创建参数的时候一般会有真实值、与显示值。真实值就是传参中的编码、ID等值；显示值就是最终用户能懂得的名称。

示例说明 从“源_电子表格”报表跳转传参到“目标_灵活分析”报表。“目标_灵活分析”的参数“产品类别”的真实值为“产品类别编码”、显示值为“产品类别名称”。

只传真实值为“产品类别编码”，如下图：

当前位置： 周艳 >> 电子表格 >> 电子表格_传真实显示值

| 产品类别编码 | 产品类别名称 | 华东 | | 华北 | | 华南 | | |
|--------|--------|------|------|------|------|------|------|-----|
| 7 | 特制品 | 506 | 924 | 1341 | 2170 | 689 | 931 | 83 |
| 8 | 海鲜 | 1371 | 1324 | 2766 | 2312 | 1647 | 1342 | 166 |
| 2 | 调味品 | 1112 | 1205 | 1952 | 1796 | 1239 | 855 | 145 |
| 5 | 谷类/麦片 | 920 | 939 | 1646 | 1476 | 1141 | 920 | 259 |
| 3 | 点心 | 1523 | 1604 | 3117 | 2845 | 1637 | 1528 | 297 |
| 4 | 日用品 | 1631 | 1977 | 3919 | 4325 | 1828 | 1787 | 360 |
| 1 | 饮料 | 1946 | 3226 | 3591 | 4278 | 2169 | 2284 | 373 |
| 6 | 肉/家禽 | 782 | 1577 | 1454 | 3161 | 771 | 1073 | 197 |

参数接收了真实值，可以进行过筛选数据，但是最终用户不能明白“产品类别编码”是什么

目标_灵活分析

产品类别 2

首页 上页 下页 尾页 第 1 页，共 22 页 每页 10 行，共 214 行

| 产品名称 | 区域 | 数量 | 单价 | 销售额 |
|------|----|----|----|--------|
| 海苔酱 | 华东 | 15 | 16 | 214.20 |
| 海苔酱 | 华北 | 20 | 16 | 336.00 |
| 辣椒粉 | 华南 | 12 | 10 | 124.80 |
| 辣椒粉 | 华南 | 15 | 10 | 156.00 |
| 麻油 | 华南 | 65 | 17 | 884.00 |
| 麻油 | 华东 | 12 | 17 | 163.20 |
| 蚝油 | 华东 | 16 | 15 | 248.00 |
| 甜辣酱 | 华东 | 8 | 35 | 280.80 |
| 味精 | 华北 | 20 | 12 | 248.00 |
| 蚝油 | 华东 | 21 | 15 | 325.50 |

即传真实值为“产品类别编码”又传显示值为“产品类别名称”，如下图：

| 产品类别编码 | 产品类别名称 | 华东 | | 华北 | | 华南 | | 东 |
|--------|--------|------|------|------|------|------|------|-----|
| 7 | 特制品 | 506 | 924 | 1341 | 2170 | 689 | 931 | 83 |
| 8 | 海鲜 | 1371 | 1324 | 2766 | 2312 | 1647 | 1342 | 166 |
| 2 | 调味品 | 1112 | 1205 | 1952 | 1796 | 1239 | 855 | 145 |
| 5 | 谷类/麦片 | 920 | 939 | 1646 | 1476 | 1141 | 920 | 259 |
| 3 | 点心 | 1523 | 1604 | 3117 | 2845 | 1637 | 1528 | 297 |
| 4 | 日用品 | 1671 | 1977 | 3919 | 4325 | 1828 | 1787 | 360 |
| 1 | 饮料 | 1946 | 3225 | 3591 | 4278 | 2169 | 2284 | 373 |
| 6 | 肉/家畜 | 1782 | 1577 | 1454 | 3161 | 771 | 1073 | 197 |

即传真实值，又传显示值。则即可以进行筛选数据，
又可以使最终用明白“产品类别”的意思

目标：灵活分析

产品类别：调味品

首页 | 上一页 | 下一页 | 尾页 | 第1 页，共22页 每页10 行，共214行

| 产品名称 | 区域 | 数量 | 单价 | 销售额 |
|------|----|----|----|--------|
| 海苔酱 | 华东 | 15 | 16 | 214.20 |
| 海苔酱 | 华北 | 20 | 16 | 336.00 |
| 辣椒粉 | 华南 | 12 | 10 | 124.80 |
| 辣椒粉 | 华南 | 15 | 10 | 156.00 |
| 麻油 | 华南 | 65 | 17 | 884.00 |
| 麻油 | 华东 | 12 | 17 | 163.20 |
| 蚝油 | 华东 | 16 | 15 | 248.00 |
| 甜辣酱 | 华东 | 8 | 35 | 280.80 |
| 味精 | 华北 | 20 | 12 | 248.00 |
| 蚝油 | 华东 | 21 | 15 | 325.50 |

设置方法

- 1、在【资源定制】创建案例类似的页面。
- 2、选中电子表格，右键选择 **编辑宏** 进入报表宏界面，右击【编辑宏】>创建【跳转规则向导】。（对面上的电子表格报表进行建跳转规则）
- 3、在双点打开已创建的跳转规则的link，在该界面中进行修改【跳转规则向导】生成宏代码，如示例上中“源_电子表格”报表的“A4”列是“产品类别编码”；“B4”列是“产品类别名称”。如下图：

名称：传参_link 类型：ServerSide 对象：spreadsheetReport 事件：onBeforeOutput

```

1 function main(spreadsheetReport) {
2     if (!spreadsheetReport) return spreadsheetReport;
3     var sheet = spreadsheetReport.getSheetByName("Sheet1");
4     var positions = "C4,D4".split(",");
5     for (var i = 0; i < positions.length; i++) {
6         var position = positions[i].trim();
7         var expandedPositions = sheet.getExpandedPositions(position);
8         for (var j = 0; j < expandedPositions.length; j++) {
9             var cell = sheet.getCell(expandedPositions[j]);
10            if (true) {
11                addLink(spreadsheetReport, sheet, cell);
12            }
13        }
14    }
15 }
16 function addLink(spreadsheetReport, sheet, cell) {
17     sheet.addRuleLink(cell.getCellPosition(), "传参", [
18         getCellValue(cell.getRelativeCells('A4')), getCellValue(cell.getRelativeCells('B4'))]); //修改的代码部份，只需再添加获取
19 }
20 function getCellValue(cell) {
21     if (!cell) return null;
22     if (typeof cell.length == 'number') {
23         var ret = [];
24         for (var i = 0; i < cell.length; i++) {
25             ret.push(cell[i].value);
26         }
27         return ret;
28     } else {
29         return cell.value;
30     }
31 }
32 function checkValue(v1, v2, op) {
33     if (v1 && typeof v1.length == 'number') {
34         for (var i = 0; i < v1.length; i++) {
35

```

参数接收的真实值 “产品类别编码”

参数接收的显示值 “产品类别名称”

- 4、在双点打开已创建的跳转规则的action，在该界面中进行修改【跳转规则向导】生成宏代码，因在第3步中修改了link，在link中添加了参数的显示值“产品类别名称”B4列，所以action中修改参数接收值之后的填充中的参数显示值为params[1]。如下图：

名称: 传参_action 类型: ClientSide 对象: spreadsheetReport 事件: onRender

```
1 function paramToString(v) {
2   return v == null ? null: v.toString();
3 }
4 function main(spreadsheetReport, ruleName, params, page, pageContext) {
5   if (ruleName != "传参") return;
6   var targetResId = "I4028818a1af1b4c2014ea92deff11cc8";
7   var newReport = false;
8   //找到对应的dashlet对象，并刷新它。
9   var dashlet = page.findDashletByReportId(targetResId);
10  //dashlet不存在
11  if (!dashlet) return;
12  var portlet = dashlet.currentPortlet;
13  if (!portlet) {
14    return;
15  }
16  if (typeof paramsInfo != "undefined" && paramsInfo) paramsInfo.push({
17    name: "产品类别_code",
18    value: paramToString(params[0]),
19    displayValue: paramToString(params[1]) //修改代码的部份只需求改参数显示值，把displayValue: paramToString(params[0])
20  });
21  var pObj = {
22    value: paramToString(params[0]),
23    displayValue: paramToString(params[1]) //修改代码的部份只需求改参数显示值，把displayValue: paramToString(params[0])
24  };
25  if (typeof portlet != "undefined" && portlet) portlet.fillParamValueFromObj("产品类别_code", pObj);
26  //设置参数值到报表对象
27  portlet.commitParamValue();
28  portlet.refreshData();
29 }
30 //创建参数信息
31 function createParamInfo(name, value, disValue, mergeType) {
32   var paramInfo = new Object();
33   paramInfo.name = name;
34   paramInfo.value = value;
35 }
```

参数真实值

参数显示值

参数真实值

参数显示值

link 宏代码

```

function main(spreadsheetReport) {
  if (!spreadsheetReport) return spreadsheetReport;
  var sheet = spreadsheetReport.getSheetByName("Sheet1");
  var positions = "C4,D4".split(",");
  for (var i = 0; i < positions.length; i++) {
    var position = positions[i].trim();
    var expandedPositions = sheet.getExpandedPositions(position);
    for (var j = 0; j < expandedPositions.length; j++) {
      var cell = sheet.getCell(expandedPositions[j]);
      if (true) {
        addLink(spreadsheetReport, sheet, cell);
      }
    }
  }
}

function addLink(spreadsheetReport, sheet, cell) {
  sheet.addRuleLink(cell.cellPosition, "", [
    getCellValue(cell.getRelativeCells('A4')), getCellValue(cell.getRelativeCells('B4'))]); //getCellValue(cell.
getRelativeCells('B4'))
}

function getCellValue(cell) {
  if (!cell) return null;
  if (typeof cell.length == 'number') {
    var ret = [];
    for (var i = 0; i < cell.length; i++) {
      ret.push(cell[i].value);
    }
    return ret;
  } else {
    return cell.value;
  }
}

function checkValue(v1, v2, op) {
  if (v1 && typeof v1.length == 'number') {
    for (var i = 0; i < v1.length; i++) {
      if (checkValue(v1[i], v2, op)) return true;
    }
    return false;
  }
  switch (op) {
    case 'EQUAL':
      return (v1 == v2) || (v2.match(/^\\d+(\\.\\d+)?$/)) && v1 == parseFloat(v2);
    case 'GREATER':
      return v1 > parseFloat(v2);
    case 'LESS':
      return v1 < parseFloat(v2);
    case 'INCLUDE':
      return v1.indexOf(v2) >= 0;
    case 'PREFIX':
      return v1.indexOf(v2) == 0;
    case 'SUFFIX':
      return v1.indexOf(v2) == (v1.length - v2.length);
  }
}

```

action宏代码

```

function paramToString(v) {
    return v == null ? null: v.toString();
}
function main(spreadsheetReport, ruleName, params, page, pageContext) {
    if (ruleName != "") return;
    var targetResId = "I4028818a1af1b4c2014ea92deff11cc8";
    var newReport = false;
    //dashlet
    var dashlet = page.findDashletByReportId(targetResId);
    //dashlet
    if (!dashlet) return;
    var portlet = dashlet.currentPortlet;
    if (!portlet) {
        return;
    }
    if (typeof paramsInfo != "undefined" && paramsInfo) paramsInfo.push({
        name: "_code",
        value: paramToString(params[0]),
        displayValue: paramToString(params[1]) //displayValue: paramToString(params[0])displayValue:
paramToString(params[1])
    });
    var pObj = {
        value: paramToString(params[0]),
        displayValue: paramToString(params[1]) //displayValue: paramToString(params[0])displayValue:
paramToString(params[1])
    };
    if (typeof portlet != "undefined" && portlet) portlet.fillParamValueFromObj("_code", pObj);
    //
    portlet.commitParamValue();
    portlet.refreshData();
}
//
function createParamInfo(name, value, disValue, mergeType) {
    var paramInfo = new Object();
    paramInfo.name = name;
    paramInfo.value = value;
    paramInfo.displayValue = disValue;
    if (mergeType) paramInfo.groupType = mergeType;
    return paramInfo;
}

```