

# Excel文件作为报表数据的过滤条件

## 1、需求背景

客户有个需求是，提供一个excel，excel的某一列或几列有数据，想办法通过已有的数据为条件，查询出其它列数据补充完整。譬如：手头已经有个excel文件，这里面是他需要关注的客户，每次查也都是这批客户，希望直接查出这批客户的其他相关信息。

**方案：**通过excel模板导入数据到数据库表中，并且只为每个用户保留他自己最后一次导入的数据，然后将报表的参数值来源于这张库表或者直接用这张表与现有数据关联查询，最后导出成excel，即为客户需要的excel。

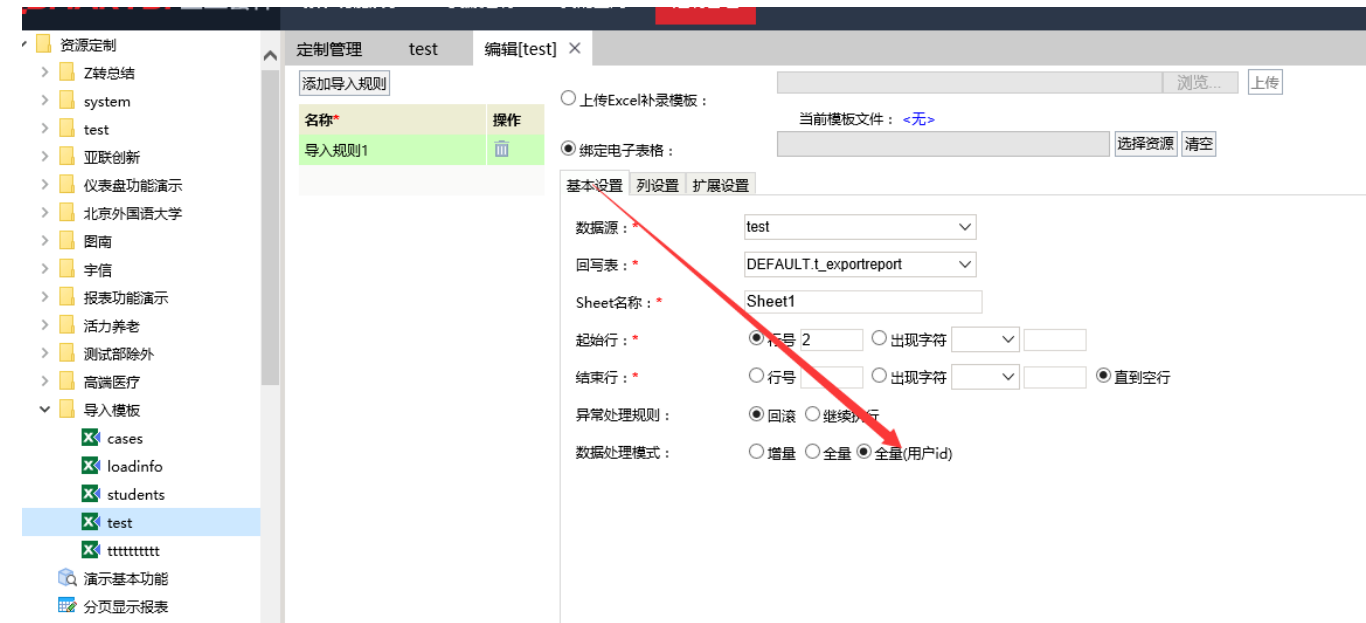
**定制点：**产品原有的excel导入功能只有全量（把以往导的全删）和增量，这里需要按用户全量，就是A用户导入excel文件时，需要删除A用户以往导入的数据，其他用户的还是要保留。所以在excel导入数据到表中时，需根据用户名进行导入。

注意：本示例来源于实际项目，原始版本是V6.1。

此示例同样也适用于 v7

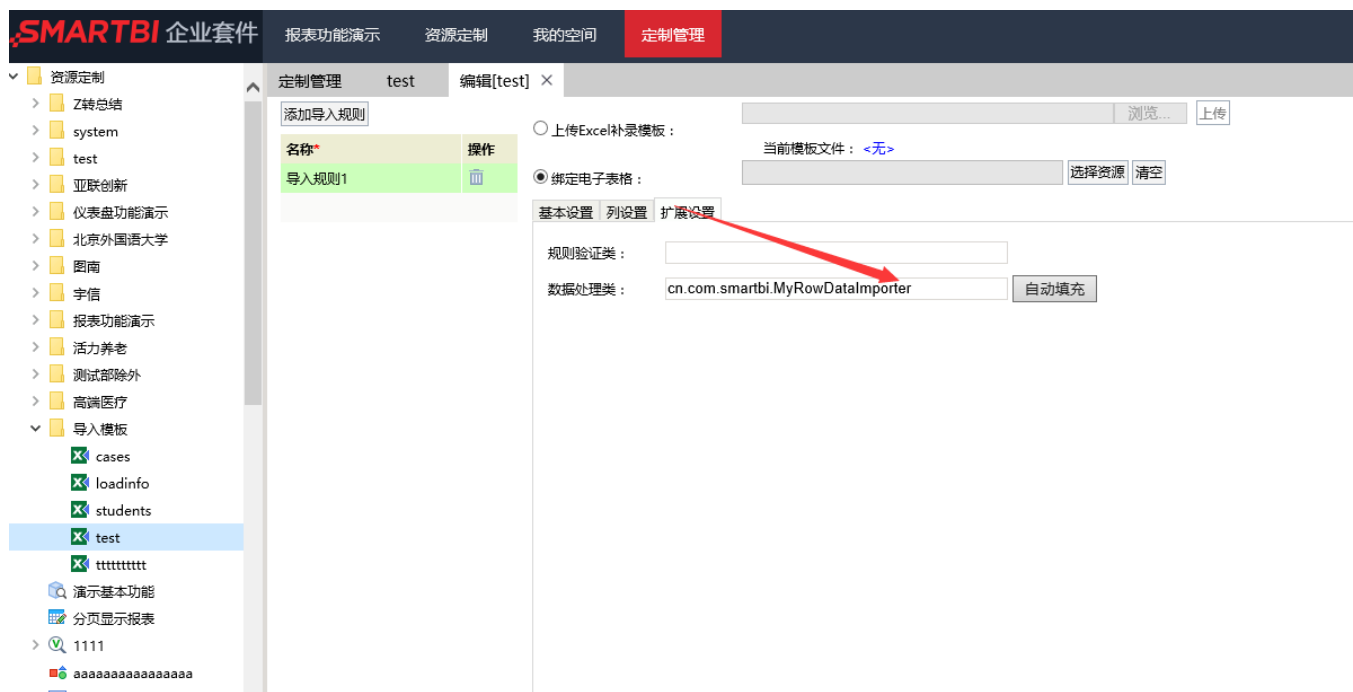
## 2、实现方案

增加数据处理模式：按当前用户全量删除



增加数据处理类，可点击后面按钮自动添加

- 1、需求背景
- 2、实现方案
- 3、主要代码
- 4、简单应用示例
- 5、如何将excel导入模版绑定到报表中
- 6、相关资源（EPPR-8801）



绑定数据表，必须有一列来存放用户名



### 3、主要代码

#### MyRowDataImporter类

```
package cn.com.smartbi;
import java.sql.Connection;
import java.sql.Statement;
import javax.servlet.http.HttpSession;
import org.apache.log4j.Logger;
import smartbi.dag.excelimport.DefaultRowDataImporter;
import smartbi.dag.excelimport.ImportFileConfig;
import smartbi.dag.excelimport.RowData;
import smartbi.dag.excelimport.RowDataImporterResult;
import smartbi.net.sf.json.JSONArray;
```

```

import smartbi.net.sf.json.JSONObject;
import smartbi.state.IStateModule;
import smartbi.state.StateModule;
import smartbi.usermanager.User;
import smartbi.usermanager.UserManagerModule;
import smartbi.util.StringUtil;
public class MyRowDataImporter extends DefaultRowDataImporter {
    private static final Logger LOG = Logger.getLogger(MyRowDataImporter.class);

    /**
     * ,
     */
    @Override
    public RowDataImporterResult process(RowData rowData, ImportFileConfig config) throws Exception {
        if(!isDeleteRows()){
            JSONObject jsonObj = config.getConfig();
            User user = (User) UserManagerModule.getInstance().getCurrentUser();
            String username = user.getName();
            String column = "";
            String dataprocessMode = jsonObj.optString("dataprocessMode");
            JSONArray json = jsonObj.optJSONArray("columnConfigs");
            for(int i=0;i<json.length();i++){
                JSONObject job = json.getJSONObject(i);
                if(job.has("javaClassName")&&"smartbi.daq.excelimport.
CurrentUserIDCellDataGenerator".equals(job.getString("javaClassName"))){
                    column = job.getString("fieldName"); //ID
                }
            }
            //
            if (StringUtil.equals("UserFull", dataprocessMode)) {
                String tableName = jsonObj.optString("tableName");
                String deleteSql = "delete from " + tableName + " where "+column+"
                = '"+username+"'";

                // String dsId = jsonObj.optString("dataSourceId");
                Connection conn = config.getConn();
                try {
                    Statement stat = conn.createStatement();
                    stat.execute(deleteSql);
                    setIsDeleteRows("true"); //
                } catch (Exception e) {
                    LOG.error(e.getMessage(), e);
                }
            }
        }
        return super.process(rowData, config);
    }

    //excel,,
    public boolean isDeleteRows(){
        IStateModule stateModule = StateModule.getInstance();
        HttpSession session = stateModule.getSession();
        String isDeleteRows = (String) session.getAttribute("isDeleteRows");
        if("true".equals(isDeleteRows)){
            return true;
        }
        return false;
    }

    public void setIsDeleteRows(String attr){
        IStateModule stateModule = StateModule.getInstance();
        HttpSession session = stateModule.getSession();
        session.setAttribute("isDeleteRows", attr);
    }
}

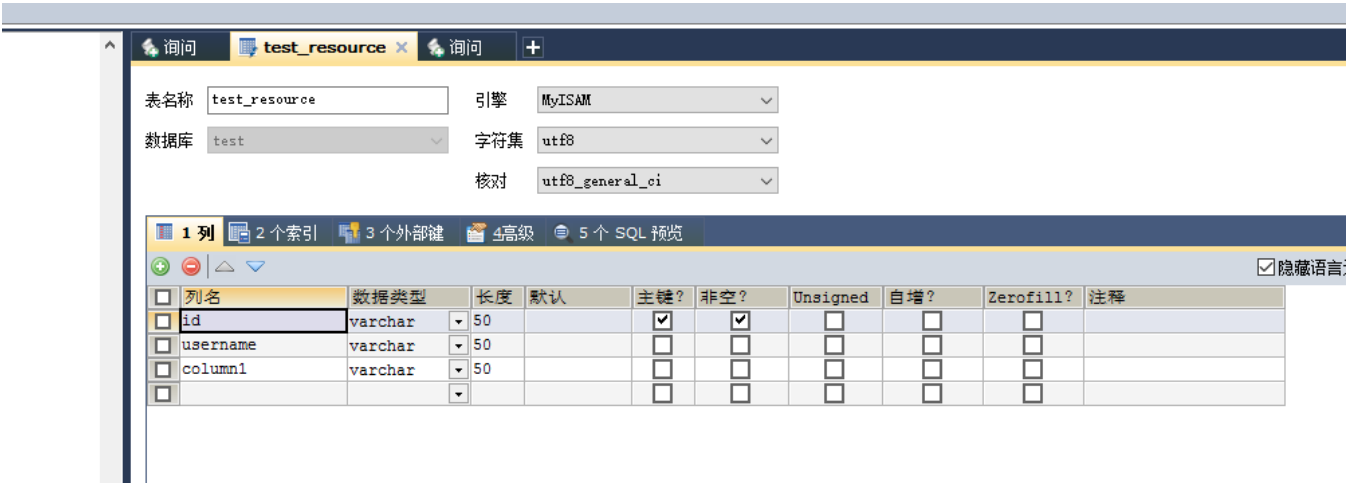
```

## 4、简单应用示例

第一步：

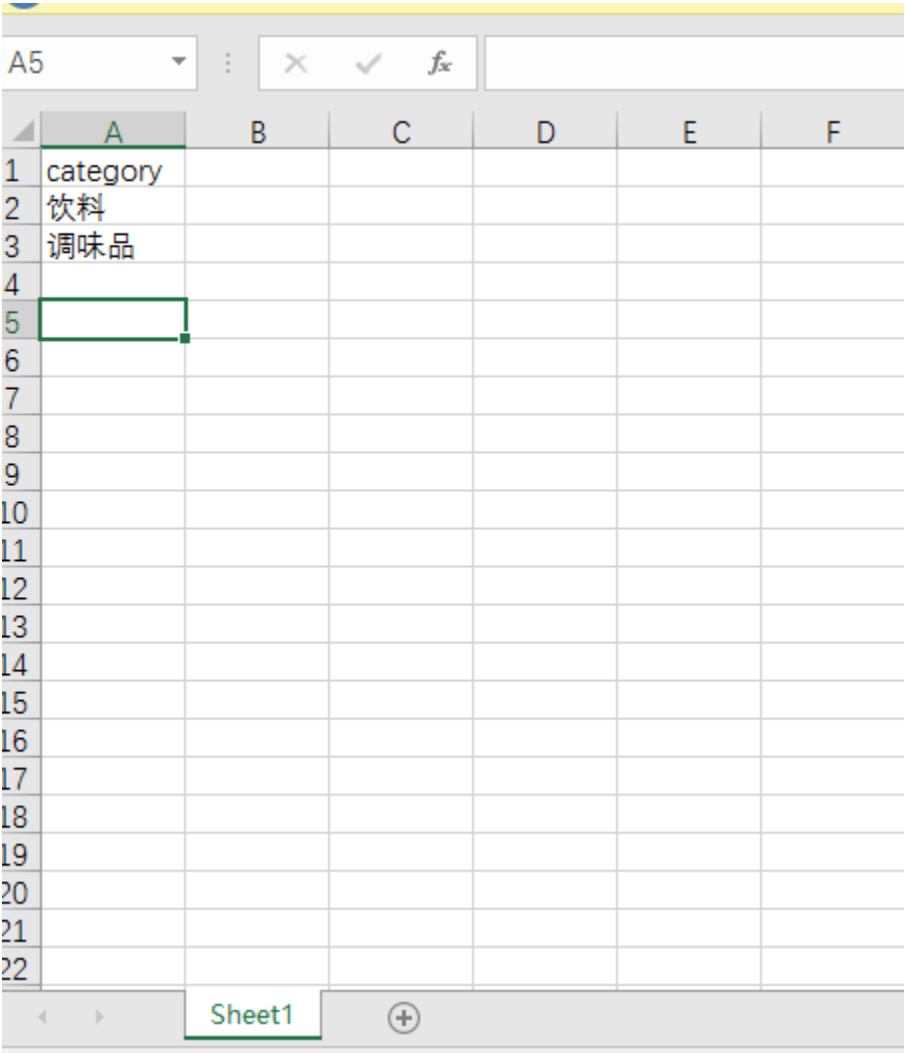
准备好一个数据表，如下图：

id为唯一标识，username存放当前用户名（必要），column1为条件数据



第二步：

将上表加入到数据源中，根据excel数据新建导入模版（参考[Excel数据导入](#)），设置如下：



定制管理

新建Excel导入模板

添加导入规则

名称\*

操作

导入规则1

上传Excel补录模板：

选择文件

未选择任何文件

上传

当前模板文件：<无>

绑定电子表格：

选择资源

清空

基本设置

列设置

扩展设置

数据源：

test

回写表：

DEFAULT.test\_resource

Sheet名称：

Sheet1

起始行：

行号

2

出现字符

结束行：

行号

出现字符

直到空行

异常处理规则：

回滚

继续执行

数据处理模式：

增量

全量

全量(用户id)

定制管理

新建Excel导入模板

添加导入规则

名称\*

操作

导入规则1

上传Excel补录模板：

选择文件

未选择任何文件

上传

当前模板文件：<无>

绑定电子表格：

选择资源

清空

基本设置

列设置

扩展设置

字段名称	字段类型	主键	允许空值	绑定输入	校验设置
id	VARCHAR(50)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	使用JAVA类	类名：唯一id
username	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	使用JAVA类	类名：当前用户名称
column1	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	绑定Excel列	列号：A

报表功能演示

资源定制

我的空间

定制管理

定制管理

新建Excel导入模板

添加导入规则

名称\*

操作

导入规则1

上传Excel补录模板：

选择文件

未选择任何文件

上传

当前模板文件：<无>

绑定电子表格：

选择资源

清空

基本设置

列设置

扩展设置

规则验证类：

数据处理类：

cn.com.smartbi.MyRowDataImporter

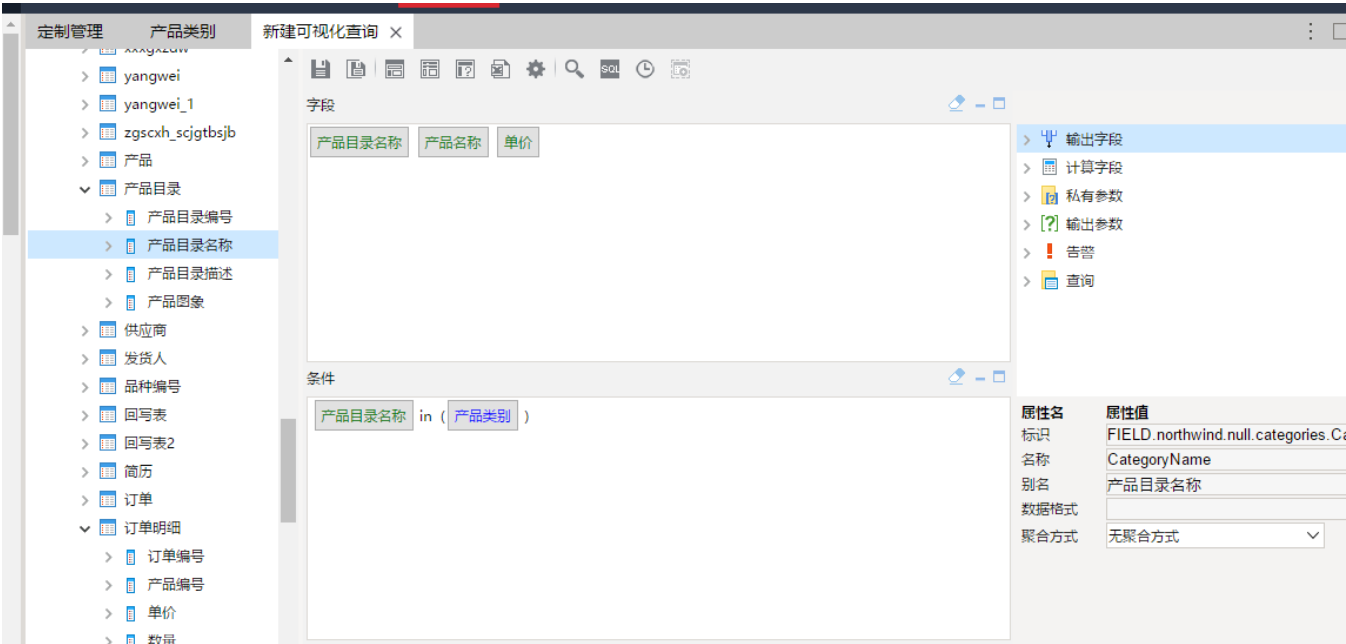
自动填充

第三步：

制作报表，首先创建参数



新建查询，引用上述参数：



制作报表，查看数据

比如我用admin用户导入两条数据（饮料，调味品）

牛 报表功能演示 资源定制 我的空间 定制管理

定制管理 test x

产品类别\* 饮料,调味品

[首页][上一页] [下一页][末页] 每页10行, 共619行

产品目录

产品目录	单价
调味品	16
调味品	16
饮料	14
饮料	3
饮料	15
调味品 辣椒粉	10
饮料 运动饮料	14
调味品 辣椒粉	10
饮料 牛奶	15
调味品 麻油	17

我用testuser用户导入两条数据（点心、日用品）

报表功能演示 资源定制 我的空间 定制管理

定制管理 测试模版 test x

产品类别\* 点心,日用品

[首页][上一页] [下一页][末页] 每页10行, 共700行

产品目录

产品目录	单价
点心	64
日用品	2
日用品	27
日用品	10
点心	16
点心 饼干	13
日用品 苏澳奶酪	44
点心 牛肉干	35
日用品 白奶酪	25
点心 花生	8

数据表中的数据如下：

id	username	column1
I402882bb015ada10da103840015ada27067f0034	admin	饮料
I402882bb015ada10da103840015ada2706810037	admin	调味品
I402882bb015ada10da103840015ada39d22300d7	testuser	点心
I402882bb015ada10da103840015ada39d22500da	testuser	日用品
*(NULL)	(NULL)	(NULL)

如果再次导入数据，会覆盖当前用户对原有数据

## 5、如何将excel导入模版绑定到报表中

如果是电子表格报表，可参考[导入模版绑定到电子表格报表](#)。

如果是其它类型报表，或者需要一个模版绑定多个报表，可参考下面示例代码：

可用[第三课：宏资源包](#)配置多个报表对应的模版

### 1、电子表格

ClientSide（客户端）      spreadsheetReport（对象）      onRender（事件）



```

var mappings = {
  // "ID
  'I2c91918a27eff3ab0145a86c6a4317e1': {
    //ID
    'importId': 'I4028828b015774ec74ec29f30157751e3c18005d'
  },
  //
  'id': {
    //ID
    'importId': 'I4028828b015774ec74ec29f30157751e3c18005d'
  }
}

function main(spreadsheetReport) {
  // debugger;
  //
  var td = spreadsheetReport.elem_space2;
  if (!spreadsheetReport._newBtn) { //
    var input = document.createElement("INPUT");
    input.type = "button";
    input.value = "";
    input.title = "";
    input.style.width = "100";
    input.style.height = "20";
    var newBtn = td.appendChild(input);
    spreadsheetReport.addListener(input, "click", doNewButtonClick,
      spreadsheetReport);
    spreadsheetReport._newBtn = newBtn;
  }
  // debugger;
  // Click
  function doNewButtonClick(e) {
    var arr = new Array();
    var obj = new Object();
    obj.id = getImportId(spreadsheetReport);
    arr.push(obj);
    if (!dialogFactory) {
      var dialogFactory = jsloader.resolve("freequery.dialog.dialogFactory");
    }
    var dialogConfig = {};
    dialogConfig.title = "";
    dialogConfig.size = dialogFactory.size.LARGE;
    dialogConfig.fullName = "smartbi.spreadsheetreport.OpenImportConfigDialog";
    dialogFactory.showDialog(dialogConfig, [arr], null, this);
  }
}

function getImportId(spreadsheetReport) {debugger;
  return mappings[spreadsheetReport.queryId].importId || "";
}

```

## 2、灵活分析

ClientSide（客户端）      simpleReport（对象）      onRender（事件）

```

var mappings = {
  // ""ID
  'I2c94ea86296db80801296dd20f12005a': {
    //ID
    'importId': 'I4028828b015774ec74ec29f30157751e3c18005d'
  },
  //
  'id': {
    //ID
    'importId': 'I4028828b015774ec74ec29f30157751e3c18005d'
  }
}

function main(simpleReport, simpleReportContext) {
  var node = simpleReport.pagePanel.firstpage.parentNode;
  //
  var queryBtn = document.createElement("button");
  queryBtn.value = ""; //
  queryBtn.innerHTML = "";
  queryBtn.className = "query-button";
  queryBtn.style.marginLeft = "10px";
  node.appendChild(queryBtn);
  simpleReport.addListener(queryBtn, "click", doNewButtonClick, this);
  function doNewButtonClick(e) {
    var arr = new Array();
    var obj = new Object();
    obj.id = getImportId(simpleReport);
    arr.push(obj);
    if (!dialogFactory) {
      var dialogFactory = jsloader.resolve("freequery.dialog.dialogFactory");
    }
    var dialogConfig = {};
    dialogConfig.title = "";
    dialogConfig.size = dialogFactory.size.LARGE;
    dialogConfig.fullName = "smartbi.spreadsheetreport.OpenImportConfigDialog";
    dialogFactory.showDialog(dialogConfig, [arr], null, this);
  }
}

function getImportId(simpleReport) {
  return mappings[simpleReport.queryId].importId || "";
}

```

### 3、组合分析

ClientSide（客户端）      simpleReport（对象）      onRenderTable（事件）

```

var mappings = {
  // ""ID
  'I2c94ea86296db80801296dd20f12005a': {
    //ID
    'importId': 'I4028828b015774ec74ec29f30157751e3c18005d'
  },
  //
  'id': {
    //ID
    'importId': 'I4028828b015774ec74ec29f30157751e3c18005d'
  }
}

function main(simpleReport, simpleReportContext) {
  var node = simpleReport.pagePanel.firstpage.parentNode;

  if(!simpleReport.newButton){
    //
    var queryBtn = document.createElement("button");
    queryBtn.value = ""; //
    queryBtn.innerHTML = "";
    queryBtn.className = "query-button";
    queryBtn.style.marginLeft = "10px";
    node.appendChild(queryBtn);
    simpleReport.newButton = queryBtn;
    simpleReport.addListener(queryBtn, "click", doNewButtonClick, this);
  }
  function doNewButtonClick(e) {
    var arr = new Array();
    var obj = new Object();
    obj.id = getImportId(simpleReport);
    arr.push(obj);
    if (!dialogFactory) {
      var dialogFactory = jsloader.resolve("freequery.dialog.dialogFactory");
    }
    var dialogConfig = {};
    dialogConfig.title = "";
    dialogConfig.size = dialogFactory.size.LARGE;
    dialogConfig.fullName = "smartbi.spreadsheetreport.OpenImportConfigDialog";
    dialogFactory.showDialog(dialogConfig, [arr], null, this);
  }
}

function getImportId(simpleReport) {
  return mappings[simpleReport.queryId].importId || "";
}

```

## 6、相关资源（EPPR-8801）

可能涉及到表不同，资源不一定能用，需稍作一定的修改。

[migrate.xml](#)

[Excelimport.rar](#)

[excelimport.ext](#)