** 深圳天源迪科信息技术股份有限公司**

**合作伙伴佣金新前台公共方法**

**使用说明**

本文件属深圳天源迪科信息技术股份有限公司

所有未经书面许可不得以任何形式复印或传播

**文件建立/修改记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 版本 | 建立或修改 | 建立/修改人  日期 | 审核人  日期 | 批准人  日期 |
| 1 | V1.0.0 | 建立 | 何异  2017-7-20 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

[Js 4](#_Toc27363)

[打开弹出框 4](#_Toc5659)

[公共函数 4](#_Toc1212)

[说明 4](#_Toc24813)

[使用示例 5](#_Toc10576)

[成功消息提示框 5](#_Toc31158)

[公共函数 5](#_Toc2693)

[错误级别消息提示框 6](#_Toc5639)

[公共函数 6](#_Toc11748)

[警告级别消息提示框 6](#_Toc6806)

[公共函数 6](#_Toc19795)

[确认提示的消息提示框 6](#_Toc18833)

[公共函数 6](#_Toc23202)

[使用示例 7](#_Toc3198)

[导出excel 7](#_Toc9325)

[公共函数 7](#_Toc32267)

[说明 7](#_Toc20256)

[使用示例 8](#_Toc15234)

[Ajax调用 8](#_Toc11050)

[公共函数 8](#_Toc700)

[使用示例 9](#_Toc31126)

[URL配置 9](#_Toc2247)

[Css 11](#_Toc31401)

[Service 11](#_Toc5785)

# Js

## 打开弹出框

### 公共函数

/\*\*

\* 页面为修改操作，则调用该方法

\* **@param** title 弹出框的标题

\* **@param** width 宽度

\* **@param** height 高度

\* **@param** url url

\* **@param** callback 回调函数

\* **@param** params 传递参数 自定义格式

\* **@param** opt 选项，自定义扩展

\* **@date** 20170719

\*/

**function** showEditDialog(title,width,height,url,callback,params,opt)

/\*\*

\* 页面为增加操作，则调用该方法

\* **@param** title 弹出框的标题

\* **@param** width 宽度

\* **@param** height 高度

\* **@param** url url

\* **@param** callback 回调函数

\* **@param** params 传递参数 自定义格式

\* **@param** opt 选项，自定义扩展

\* **@date** 20170719

\*/

**function** showAddDialog(title,width,height,url,callback,params,opt)

### 说明

这两个函数唯一的区别在于onLoadComplete函数的第一个参数，调用这两个函数打开子页面之后，在子页面定义onLoadComplete函数，action参数分别获取到的是edit和add

### 使用示例

**父页面调用如下：**

**var** addObj = rows[0];

addObj['oper\_flag'] = 'A';

addObj['service\_name'] = 'REF\_DEM\_TYPE\_OFR\_INSERT';

showAddDialog(

"套餐选择 ",600,400,Globals.baseHtmlUrl.OFFER\_CLASS\_ADD\_OFFER\_URL,

**function** callback(data){

**if** (data != **undefined** && data.success == systemVar.SUCCESS) {

queryOffer(addObj["REF\_DEM\_TYPE\_ID"]);

}**else** **if**(data == systemVar.CANCEL || data == systemVar.CLOSE){

}**else**{

showErrorMessageAlter(data.errormsg);

**return** **false**;

}

},addObj);

子页面接收父页面参数

/\*\*

\* 父窗口调用的初始化方法

\* **@param** action 操作类型

\* **@param** data 参数

\*/

**function** onLoadComplete(action,data) {

JsVar["classForm"].setData(data,**false**);

//父窗口参数

JsVar["data"] = data;

}

## 成功消息提示框

### 公共函数

/\*\*

\* 显示成功级别的消息弹出框

\* **@param** msg 消息内容 必输项

\* **@param** callback 回调函数 可选项

\* **@date** 20170719

\*/

**function** showMessageAlter(msg,callback)

## 错误级别消息提示框

### 公共函数

/\*\*

\* 显示error级别的消息弹出框

\* **@param** msg 消息内容 必输项

\* **@param** callback 回调函数 可选项

\*/

**function** showErrorMessageAlter(msg,callback)

## 警告级别消息提示框

### 公共函数

/\*\*

\* 显示warning级别的消息弹出框

\* **@param** msg 消息内容 必输项

\* **@param** callback 回调函数 可选项

\*/

**function** showWarnMessageAlter(msg,callback)

## 确认提示的消息提示框

### 公共函数

/\*\*

\* 显示确定提示的消息弹出框

\* **@param** msg 消息内容 必输项

\* **@param** okcallback 确定回调函数 必输项

\* **@param** cancelcallback 取消回调函数 可选项

\* **@date** 20170719

\*/

**function** showConfirmMessageAlter(msg,okcallback,cancelcallback)

### 使用示例

showConfirmMessageAlter("确定删除套餐分类？",**function** ok(){

getJsonDataByPost(Globals.baseActionUrl.ACTION\_OPERATION\_URL,delObj,"套餐分类管理-删除套餐分类",

**function**(result){

**if** (result.success == systemVar.SUCCESS) {

showMessageAlter("删除套餐分类成功!");

queryData();

}

**else**

{

showErrorMessageAlter(result.errormsg);

}

},**null**);

});

## 导出excel

### 公共函数

/\*\*

\* 导出excel

\* **@param** url 生成excel内容的url

\* **@param** columns 导出datagrid的列

\* **@param** params 生成excel内容需要的参数

\*/

**function** exportExcel(url,columns,params)

### 说明

params是一个Object，包含生成excel内容调用的service\_name，生成excel文件名称file\_name，执行service\_name需要的一些参数

会导出datagrid内除type=*"checkcolumn"* 的列，写excel内容时，程序会按照datagrid内field的width属性除以10之后的值作为输出excel的列宽，如果没有设置width，则默认宽度为25

### 使用示例

/\*\*

\* 导出套餐

\*/

**function** exportOffer(){

mini.parse();

**var** rows = mini.get("class\_grid").getSelecteds();

**if**(rows.length != 1){

showErrorMessageAlter("请选中一个套餐分类导出关联套餐!");

**return**;

}

**var** grid = mini.get("offer\_rel\_grid");

**var** params = **new** Object();

params["service\_name"] = "REF\_DEM\_TYPE\_OFR\_EXPORT";

params["file\_name"] = rows[0]["REF\_DEM\_TYPE\_NAME"]+"关联套餐";

params["ref\_dem\_type\_id"] = rows[0]["REF\_DEM\_TYPE\_ID"];

exportExcel(Globals.baseActionUrl.EXPORT\_OPERATION\_URL,grid.columns,params);

}

## Ajax调用

### 公共函数

/\*\*

\* 通过post方式获取json数据

\* **@param** url 地址 格式："/userAction.do?method=insertEmpee"

\* **@param** data data 参数 json格式 {id：1，name:"adc"} 或 [{id：1，name:"adc"}]

\* **@param** logName 模块名 例如：“权限管理-增加权限”

\* **@param** callback 回调函数，返回成功后回调方法

\* **@param** execKey execKey

\* **@param** dbKey dbKey 可选项

\* **@param** async 是否异步，默认异步 可选项

\* **@date** 20170718

\*/

**function** getJsonDataByPost(url,data,logName,callback,execKey,dbKey,async,msg,isShowLoadMask)

### 使用示例

getJsonDataByPost(Globals.baseActionUrl.ACTION\_OPERATION\_URL,delObj,"套餐分类管理-删除套餐分类",

**function**(result){

**if** (result.success == systemVar.SUCCESS) {

showMessageAlter("删除套餐分类成功!");

queryData();

}

**else**

{

showErrorMessageAlter(result.errormsg);

}

},**null**);

## 生成datagrid的columns

### 公共函数

/\*\*

\* 生成datagrid field

\* **@param** id datagrid的id

\* **@param** serviceName datagrid数据加载SQL

\* **@param** isMul 首列是否显示复选框

\*/

**function** initTableHead(id,serviceName,isMul)

### 使用示例

1、Html中定义datagrid

<div id=*"class\_grid"* class=*"mini-datagrid"* style="width: *100%*; height: *245px*;" allowAlternating=*"true"* idField=*"REF\_DEM\_TYPE\_ID"* allowResize=*"false"* allowCellselect=*"false"* multiSelect=*"true"*

showFooter=*"true"* sortMode=*"client"* onselectionchanged=*"onSelectionChanged"* pageSize=*"6"*>

</div>

2、调用公共js生成datagrid的columns

//初始化

$(document).ready(**function**() {

mini.parse();

//生成datagrid的field

initTableHead("class\_grid","REF\_DEM\_TYPE\_SELECT",**true**);

//生成datagrid的field

initTableHead("offer\_rel\_grid","REF\_DEM\_TYPE\_OFR\_SELECT",**true**);

});

3、配置par\_service\_fields表

insert into par\_service\_fields (SERVICE\_NAME, COL\_NAME, COL\_LEN, COL\_IDX, COL\_CODE)

values ('REF\_DEM\_TYPE\_SELECT', '分类ID', 100, 1, 'REF\_DEM\_TYPE\_ID');

insert into par\_service\_fields (SERVICE\_NAME, COL\_NAME, COL\_LEN, COL\_IDX, COL\_CODE)

values ('REF\_DEM\_TYPE\_SELECT', '分类名称', 250, 2, 'REF\_DEM\_TYPE\_NAME');

## 上传文件

### 公共函数

/\*\*

\* ajax上传文件

\* **@param** url 上传文件调用的url

\* **@param** elementID html中type定义为file的input元素ID

\* **@param** data 上传文件需要的额外参数 参数 json格式 {id：1，name:"adc"}

\* **@param** callback 上传成功后的回调函数

\*/

**function** ajaxUploadFile(url,elementID,data,callback)

### 说明

Html文件中要定义enctype=*"multipart/form-data"的form和一个type为file的input元素*

### 使用示例

Html代码

<form id=*"upload"* method=*"post"* enctype=*"multipart/form-data"*>

<table id=*"class\_form"* class=*"formTable6"* style="table-layout: *fixed*;">

<tr>

<th><span class=*"fred"*>\*</span>Excel文件：</th>

<td colspan=*"3"*><input type=*"file"* class=*"mini-htmlfile"* name=*"file\_path"* id=*"file\_path"* style="width:*300px*;"/>

</td>

</tr>

<tr>

<td colspan=*"4"*><a href=*"../../../../../template/ref\_dem\_type\_ofr.xls"* target=*"\_blank"* style="padding-left:*20px*;padding-right:*50px*;">模板下载</a> 右键另存为xls格式</td>

</tr>

</table>

<div class=*"mini-toolbar"* style="height:*28px*;text-align: *center*; padding-top: *8px*; padding-bottom:*8px*;"

borderStyle=*"border:0;border-top:solid 1px #b1c3e0;"*>

<a class=*"mini-button"* onclick="onSubmit()" id=*"add-dir"* data-options=*'{"pId":"add-dir"}'*>保存</a>

<span style="display: *inline-block*; width: *25px*;"></span>

<a class=*"mini-button"* onclick="closeWindow(systemVar.CANCEL)">取消</a>

</div>

</form>

Js代码

/\*\*

\* 导入套餐

\*/

**function** onSubmit() {

//导入套餐

**var** offerInfo = **new** Object();

offerInfo["service\_name"] = JsVar["data"]["service\_name"];

offerInfo["service\_name\_pre"] = JsVar["data"]["service\_name\_pre"];

offerInfo["file\_path"] = mini.get("file\_path").getValue();

offerInfo["REF\_DEM\_TYPE\_ID"] = JsVar["data"]["REF\_DEM\_TYPE\_ID"];

ajaxUploadFile(Globals.baseActionUrl.IMPORT\_OPERATION\_URL,"file\_path",offerInfo,**function**(result){

result = mini.decode(result);

offerInfo["success"] = result.success;

offerInfo["errormsg"] = result.errormsg;

closeWindow(offerInfo);

});

}

## URL配置

建议url配置在config.js文件中，action定义放在Globals.baseActionUrl中，html定义放在Globals.baseHtmlUrl中

//公共变量

**var** nowurl\_js=document.location.href;

nowurl\_js=nowurl\_js.substring(0,nowurl\_js.indexOf("html"));

**var** Globals = {

ctx: nowurl\_js

};

//用于action url

Globals.baseActionUrl = {

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*套餐分类维护 start\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//增删改操作

ACTION\_OPERATION\_URL : Globals.ctx +"/ControlServlet.do?serviceName=commonServlet&methodName=operation",

//分页查询操作

SELECT\_OPERATION\_URL : Globals.ctx +"/ControlServlet.do?serviceName=commonServlet&methodName=gridValue\_rows",

//导出操作

EXPORT\_OPERATION\_URL : Globals.ctx +"/ControlServlet.do?serviceName=commonServlet&methodName=exportExcel",

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*套餐分类维护 end\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

};

//用于html url

Globals.baseHtmlUrl = {

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*套餐分类维护 start\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//销售品分类-新增分类页面

OFFER\_CLASS\_ADD\_URL:Globals.ctx + "html/reward/dataManage/dimensionClass/offerClass/add.html",

//套餐选择页面

OFFER\_CLASS\_ADD\_OFFER\_URL:Globals.ctx + "html/reward/dataManage/dimensionClass/offerClass/offerSelect.html",

//套餐导入页面

OFFER\_CLASS\_IMPORT\_OFFER\_URL:Globals.ctx + "html/reward/dataManage/dimensionClass/offerClass/offerImport.html",

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*套餐分类维护 end\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*终端分类维护 start\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//终端分类-新增分类页面

TERMINAL\_CLASS\_ADD\_URL:Globals.ctx + "html/reward/dataManage/dimensionClass/terminalClass/add.html",

//终端选择页面

TERMINAL\_CLASS\_ADD\_TERMINAL\_URL:Globals.ctx + "html/reward/dataManage/dimensionClass/terminalClass/terminalSelect.html",

//终端导入页面

TERMINAL\_CLASS\_IMPORT\_TERMINAL\_URL:Globals.ctx + "html/reward/dataManage/dimensionClass/terminalClass/terminalImport.html",

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*终端分类维护 end\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

};

# Css

form.css文件重定义的表格样式，各表格样式有细微差别，使用的时候可以对比效果选择

# Service

## Grid分页查询

### 公共函数

/\*\*

\* 分页一览数据获取(供miniui使用)

\* **@throws** IOException

\*/

**public** **void** gridValue\_rows(HttpServletRequest request, HttpServletResponse response, Map<String, Object> rtnMap)

### 参数说明

service\_name：程序根据service\_name查询待展示数据，数据库中要有service\_name\_C配置，否则取不到记录数，会抛异常

SQL查询列名要和grid中field保持一致，否则无法展示

### 使用示例

//用于action url

Globals.baseActionUrl = {

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*套餐分类维护 start\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//增删改操作

ACTION\_OPERATION\_URL : Globals.ctx +"/ControlServlet.do?serviceName=commonServlet&methodName=operation",

//分页查询操作

SELECT\_OPERATION\_URL : Globals.ctx +"/ControlServlet.do?serviceName=commonServlet&methodName=gridValue\_rows",

//导出操作

EXPORT\_OPERATION\_URL : Globals.ctx +"/ControlServlet.do?serviceName=commonServlet&methodName=exportExcel",

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*套餐分类维护 end\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

};

/\*\*

\* 查询套餐分类

\*/

**function** queryData(){

mini.parse();

**var** param = {"latn\_id":mini.get("LATN\_ID").getValue(),

"ref\_type":1,

"service\_name":"REF\_DEM\_TYPE\_SELECT"};

**var** url = Globals.baseActionUrl.SELECT\_OPERATION\_URL;

**var** grid = mini.get("class\_grid");

grid.setUrl(url);

grid.load(param);

}

## ComboBox下拉数据加载

### 公共函数

/\*\*

\* 非分页一览数据获取(供miniui使用)

\* **@throws** IOException

\*/

**public** **void** gridValue(HttpServletRequest request, HttpServletResponse response, Map<String, Object> rtnMap)

### 参数说明

service\_name：程序根据service\_name查询待展示数据

SQL查询结果为两列，且查询列名和combobox的valueField、textField保持一致，否则无法展示

### 使用示例

//初始化本地网下拉列表

**var** obj = mini.get("LATN\_ID");

**var** url = nowurl+'ControlServlet.do?serviceName=commonServlet&methodName=gridValue&service\_name=LATN';

obj.load(url);

## 简单增删改操作

### 公共函数

/\*\*

\* 没有业务检查的增删改操作

\* **@throws** IOException

\*/

**public** **void** operation(HttpServletRequest request, HttpServletResponse response, Map<String, Object> rtnMap)

### 参数说明

此方法为简单的数据库增删改操作，不做任何业务检查

oper\_flag：操作类型 A表示新增，U表示修改，D表示删除

service\_name：增删改执行的SQL名称

### 使用示例

Globals.baseActionUrl对象中定义url

//增删改操作

ACTION\_OPERATION\_URL : Globals.ctx +"/ControlServlet.do?serviceName=commonServlet&methodName=operation"

调用如下：

//分类类型

classInfo["REF\_TYPE"] = 1;

classInfo["service\_name"] = JsVar["data"]["service\_name"];

classInfo["oper\_flag"] = JsVar["data"]["oper\_flag"];

classInfo["REF\_DEM\_TYPE\_ID"] = JsVar["data"]["REF\_DEM\_TYPE\_ID"];

getJsonDataByPost(Globals.baseActionUrl.ACTION\_OPERATION\_URL,classInfo,"销售品分类维护",**function**(result){

classInfo["success"] = result.success;

classInfo["errormsg"] = result.errormsg;

closeWindow(classInfo);

});

## 导出excel

### 公共函数

/\*\*

\* 导出excel

\* **@param** request

\* **@param** response

\* **@throws** Exception

\*/

**public** **void** exportExcel(HttpServletRequest request, HttpServletResponse response,Map<String, Object> rtnMap) **throws** Exception

### 参数说明

service\_name：程序根据此参数查询待导出数据

file\_name：导出excel文件名，没传，导出默认为grid.xls

### 使用示例

/\*\*

\* 导出套餐

\*/

**function** exportOffer(){

mini.parse();

**var** rows = mini.get("class\_grid").getSelecteds();

**if**(rows.length != 1){

showErrorMessageAlter("请选中一个套餐分类导出关联套餐!");

**return**;

}

**var** grid = mini.get("offer\_rel\_grid");

**var** params = **new** Object();

params["service\_name"] = "REF\_DEM\_TYPE\_OFR\_EXPORT";

params["file\_name"] = rows[0]["REF\_DEM\_TYPE\_NAME"]+"关联套餐";

params["ref\_dem\_type\_id"] = rows[0]["REF\_DEM\_TYPE\_ID"];

exportExcel(Globals.baseActionUrl.EXPORT\_OPERATION\_URL,grid.columns,params);

}

## 导入excel内容

### 公共函数

http://localhost:8080/TPSS\_SR/ImportServlet.do

### 参数说明

service\_name：导入数据插入数据库执行的SQL

service\_name\_pre：在数据入库之前执行的SQL，传入的是个json数组，包含要执行的service\_name和执行方式，没有则不传，例如：

**var** serviceArr = **new** Array();

**var** serviceObj = **new** Object();

serviceObj["service\_name"] = "REF\_DEM\_TYPE\_OFR\_IMPORT\_PRE";

serviceObj["oper\_flag"] = "U";

serviceArr[0] = serviceObj;

addObj['service\_name\_pre'] = serviceArr;

导入时需要在表PAR\_IMPORT\_COLUMN配置导入数据对应的key，修改表PAR\_IMPORT\_COLUMN配置需要重启工程

service\_name：配置导入数据入库时执行的SERVICE\_NAME

col\_no：配置导入列顺序，要和导入模板中的数据顺序保持一致

col\_name：配置数据入库的key

## 数据校验

### 公共函数

对多条记录的数据校验

/\*\*

\* 校验数据源

\* **@param** service\_name 待校验数据源操作数据库的service\_name

\* **@param** checkFlag 校验标记，excel：excel导入校验 ，other：其它校验

\* **@param** dataList 校验数据

\* **@param** errorMsg 返回的校验错误信息

\* **@throws** IllegalAccessException

\* **@throws** IllegalArgumentException

\* **@throws** InvocationTargetException

\* **@throws** NoSuchMethodException

\* **@throws** SecurityException

\*/

**public** **static** **void** check(String service\_name,String checkFlag,List<Map<String,Object>> dataList,List<String> errorMsg) **throws** IllegalAccessException, IllegalArgumentException, InvocationTargetException, NoSuchMethodException, SecurityException

对单条数据的校验

/\*\*

\* 校验数据源

\* **@param** service\_name 待校验数据源操作数据库的service\_name

\* **@param** dataMap 待校验的单条数据

\* **@param** errorMsg 返回的校验错误信息

\* **@throws** IllegalAccessException

\* **@throws** IllegalArgumentException

\* **@throws** InvocationTargetException

\* **@throws** NoSuchMethodException

\* **@throws** SecurityException

\*/

**public** **static** **void** checkOne(String service\_name,Map<String,Object> dataMap,List<String> errorMsg) **throws** IllegalAccessException, IllegalArgumentException, InvocationTargetException, NoSuchMethodException, SecurityException{

### 说明

校验规则需要在表PAR\_VALID\_RULE中配置，修改表PAR\_VALID\_RULE需要重启工程

service\_name：校验的数据库操作service\_name

data\_key：校验的列在表PAR\_IMPORT\_COLUMN中配置的col\_name，check\_func为isExists时，此列可不配，校验由check\_service\_name决定

check\_func：校验函数 校验非空：isNull 校验是否整数：isNumeric 执行SQL校验某字段是否存在：isExists，其他校验需要在CheckUtil类中添加

check\_service\_name：校验执行的service\_name

result\_flag：check\_func执行的结果和此值比对 true/false

errormsg：错误信息提示

### 使用示例

//校验excel导入数据

CheckUtil.*check*(serviceName, "excel", dataList, errorMsgs);

**if**(errorMsgs.size() > 0){

String errorTip = "";

**for**(String errorMsg:errorMsgs){

errorTip += errorMsg+"</br>";

}

rtnMap.put("success","fail") ;

rtnMap.put("errormsg",errorTip) ;

}

//删除校验

CheckUtil.*checkOne*(serviceName, params, errorMsgs);

**if** (errorMsgs.size() > 0) {

logger.error("serviceName 删除校验未通过!");

String errorTip = "";

**for** (String errorMsg : errorMsgs) {

errorTip += errorMsg + "</br>";

}

rtnMap.put("success", "fail");

rtnMap.put("errormsg", errorTip);

**return**;

}