add

1.add函数来源于在frame.jsp引入的js文件：common.js

2.我们来看看common.js中的add

/\*\*

\* 添加数据

\*

\* @param me

\* 事件源

\* @param type

\* 类型 table : 表格内新增 form:跳转表单 dialog:弹窗增加 popover:弹出遮罩层

\* @param title

\* 标题

\* @param params

\* 额外的参数 key1:v1,key2:v2

\* @param config

\*/

function add(me, type, title, params, config) {

beforeAdd(me, type);

var $this = $(me);

var dataUrl = $this.attr("data-url");

var dataId = $this.attr("data-id");

var height = $this.attr("data-height");

if (dataUrl.length < 0) {

return;

}

var \_config = {

size: BootstrapDialog.SIZE\_WIDE,

onshow: function (dialogRef) {

if (height) {

dialogRef.getModalBody().css({

overflowY: "auto",

height: height

})

}

},

onhide: function (dialogRef) {

if (dataId) {

var $dt = $("#" + dataId).DataTable();

if ($dt) {

$dt.ajax.reload();

}

}

}

};

if (config) {

$.extend(true, \_config, config);

}

if (undefined != params) {

var paramsAry = params.split(",");

if (dataUrl.indexOf("?") < 0) {

dataUrl += "?1=1";

}

$.each(paramsAry, function (idx, param) {

var paramAry = param.split(":");

dataUrl += "&" + paramAry[0] + "=" + paramAry[1];

});

}

**if (type == "table") {**

**var $table = $("#" + dataId);**

**//如果add调用指定table类型，那么将调用addRow方法**

**DataTablesUtil.data.addRow($table, dataUrl);**

} else if ("form" == type) {

redirectDataUrl(me);

} else if ("dialog" == type) {

BootstrapDialog.show({

title: title,

message: $('<div></div>').load(BOMF.CONST.WEB\_APP\_NAME + dataUrl),

size: \_config.size,

onshow: \_config.onshow,

onhide: \_config.onhide

});

} else if ("popover" == type) {

dsspMenu.loadPopOverApp(title, dataUrl);

}

}

1）我们姑且先来分析一下DataTablesUtil.data.addRow:

/\*\*

\* 行编辑新增行

\* @param $table

\* @param dataUrl

\* @param datas

\* @param params 额外参数

\*/

addRow: function ($table, dataUrl, datas, params) {

if (dataUrl) {

var index1 = dataUrl.indexOf("/rest");

if (index1 < 0)

**dataUrl = "/rest" + dataUrl;**

$.ajax({

**url: BOMF.CONST.WEB\_APP\_NAME + dataUrl,**

data: params,

dataType: "json",

success: function (data) {

DataTablesUtil.data.**\_addRow**(data, $table);

}

});

//datas是自定义的数据

} else if (datas) {

DataTablesUtil.data.\_addRow(datas, $table);

}

}

我们可以看到，在url的处理上，会先看看有没有加上/rest，没有就补上，然后用jQuery的ajax来异步调用补充一些东西：其中又为url加上了项目应用的名称（比如/xiepei）

可见，一个完整的数据源路径必须包含：/项目应用名+/rest

Q：那么url是怎样被读取的呢？

A：是通过jQuery的Ajax方法异步加载的，url指定数据源，data存放数据

2）于是，我们来看这个**$.Ajax()**，参数依次是url、data、datatype、success

查看jQuery的API我们知道：

（1）**url**

类型：String

默认值: 当前页地址。

发送请求的地址。

（2）**data**

类型：String

**发送到服务器的数据**。

**将自动转换为请求字符串格式**。

GET 请求中将附加在 URL 后。

必须为 Key/Value 格式。

如果为数组，jQuery 将自动为不同值对应同一个名称。如 {foo:["bar1", "bar2"]} 转换为 '&foo=bar1&foo=bar2'。

（3）**dataType**:   
要求为String类型的参数，预期服务器返回的数据类型。如果不指定，JQuery将自动根据http包mime信息返回responseXML或responseText，并作为回调函数参数传递。可用的类型如下：  
xml：返回XML文档，可用JQuery处理。  
html：返回纯文本HTML信息；包含的script标签会在插入DOM时执行。  
script：返回纯文本JavaScript代码。不会自动缓存结果。除非设置了cache参数。注意在远程请求时（不在同一个域下），所有post请求都将转为get请求。  
json：返回JSON数据。  
jsonp：JSONP格式。使用SONP形式调用函数时，例如myurl?callback=?，JQuery将自动替换后一个“?”为正确的函数名，以执行回调函数。  
text：返回纯文本字符串。

（4）**success**：要求为Function类型的参数，请求成功后调用的回调函数，**有两个参数**。  
         (1)由服务器返回，并根据dataType参数进行处理后的数据。  
         (2)描述状态的字符串。  
         function(data, textStatus){  
            //data可能是xmlDoc、jsonObj、html、text等等  
            this;  //调用本次ajax请求时传递的options参数  
         }

也就是说，

（1）我们的$.Ajax异步请求url指定了数据源，

（2）data指定的额外参数为空，

（3）指定响应格式为json，

（4）请求成功后触发回调函数success，其中data参数跟上面的data无关，是success的形参，将接受服务器传回的数据。在success中把data参数引用交给了\_addRow函数，这个才做实际的行添加。

下面我们就来看看\_addRow：

/\*\*

\* 私有:表格新增行数据

\* @param datas

\* @param $table

\* @private

\*/

\_addRow: function (datas, $table) {

var $dt = $table.data("dataTablesDT");

if ($.isArray(datas)) {

$.each(datas, function (idx, data) {

var tmpRow = DataTablesUtil.data.\_addSingleRow(data);

DataTablesUtil.data.putTatbleData(DataTablesUtil.const.DATA\_STORE\_NEW, $table, tmpRow);

});

} else {

var row = DataTablesUtil.data.\_addSingleRow(datas);

DataTablesUtil.data.putTatbleData(DataTablesUtil.const.DATA\_STORE\_NEW, $table, row);

}

$dt.draw();

}

层层封装，要吐了，日后再说。