

# E9表单建模前端接口API

卡片form的接口在前半部分

查询列表的接口在后半部分

## 1.说明

统一封装在全局对象window.ModeForm中

E9采用单页模式，代码推荐使用API接口操作，jQuery操作不推荐，原生JS操作DOM禁止

## 2.常用基础操作接口(适用表单所有字段类型)

### 2.1 将字段名称转换成字段id

灵活运用此方法，可实现多表单、多环境，代码块通用；解耦代码块中指定fieldid

convertFieldNameTold: function(fieldname,symbol,prefix)

#### 参数说明

| 参数        | 参数类型    | 必须 | 说明                                    |
|-----------|---------|----|---------------------------------------|
| fieldname | String  | 是  | 字段名称                                  |
| symbol    | String  | 否  | 表单标示，主表(main)/具体明细表(detail_1),默认为main |
| prefix    | Boolean | 否  | 返回值是否需要 field 字符串前缀，默认为true           |

样例

```
var fieldid = ModeForm.convertFieldNameToId("zs");
var fieldid = ModeForm.convertFieldNameToId("zs_mx", "detail_1");
var fieldid = ModeForm.convertFieldNameToId("zs_mx", "detail_1", false);
```

### 2.2 获取单子字段值

```
getFieldValue: function(fieldMark)
```

### 参数说明

| 参数        | 参数类型   | 必须 | 说明                             |
|-----------|--------|----|--------------------------------|
| fieldMark | String | 是  | 字段标示，格式 field\${字段ID}_\${明细行号} |

样例

```
var fieldvalue = ModeForm.getFieldValue("field110");
```

## 2.3 修改单个字段值

此方法修改的字段如果涉及到触发联动等，修改完会自动触发联动

```
changeFieldValue: function(fieldMark, valueInfo)
```

### 参数说明

| 参数        | 参数类型   | 必须 | 说明  |
|-----------|--------|----|---|
| fieldMark | String | 是  | 字段标示，格式 field\${字段ID}_\${明细行号}  |
| valueInfo | JSON   | 是  | 字段值信息，非浏览按钮字段格式为{value:"修改的值"};specialobj为浏览按钮信息，数组格式;showhtml属性只在文本字段类型且只读情况下生效; |

样例

```
ModeForm.changeFieldValue("field11_2", {
  value: "2,3",
  specialobj:[
    {id:"2",name:"张三"},
    {id:"3",name:"李四"}
  ]
}); //修改浏览框字段的值

ModeForm.changeFieldValue("field123", {
  value: "0.23",
  showhtml: "23%"
}); //修改文本框的值，真实值和显示值不同
```

## 2.4 改变单个字段显示属性(只读/必填等)

```
changeFieldAttr: function(fieldMark, viewAttr)
```

### 参数说明

| 参数        | 参数类型   | 必须 | 说明  |
|-----------|--------|----|---|
| fieldMark | String | 是  | 字段标示, 格式 field\${字段ID}_\${明细行号}                                   |
| viewAttr  | int    | 是  | 改变字段的状态, 1: 只读, 2: 可编辑, 3: 必填, 4: 隐藏字段标签及内容,5:隐藏行, 注意目前只有主表有隐藏行功能 |

### 样例

```
ModeForm.changeFieldAttr("field110", 1); // 字段修改为只读
```

## 2.5 同时修改字段的值及显示属性

```
changeSingleField: function(fieldMark, valueInfo, variableInfo)
```

### 参数说明

| 参数           | 参数类型   | 必须 | 说明                                 |
|--------------|--------|----|------------------------------------|
| fieldMark    | String | 是  | 字段标示, 格式 field\${字段ID}_\${明细行号}    |
| valueInfo    | JSON   | 否  | 字段值信息, 与接口2格式一致, 例: {value:"修改的值"} |
| variableInfo | JSON   | 否  | 变更属性, 例: {viewAttr:3}              |

### 样例

```
ModeForm.changeSingleField("field110", {value:"修改的值"},  
{viewAttr:"1"}); // 修改值同时置为只读
```

## 2.6 批量修改字段值或显示属性

```
changeMoreField: function(changeDatas, changeVariable)
```

## 参数说明

| 参数             | 参数类型 | 必须 | 说明          |
|----------------|------|----|-------------|
| changeDatas    | JSON | 是  | 修改的字段值信息集合  |
| changeVariable | JSON | 否  | 修改的字段显示属性集合 |

样例

```
ModeForm.changeMoreField({
  field110:{value:"修改后的值"},
  field111:{value:"2,3",
  specialobj:[
    {id:"2",name:"张三"},{id:"3",name:"李四"}
  ]},
  ...
},{
  field110:{viewAttr:2},
  field111:{viewAttr:3},
  ...
});
```

## 3.明细操作相关接口

### 3.1 添加明细及设置初始值

```
addDetailRow: function(detailMark, initAddRowData={})
```

样例

```
ModeForm.addDetailRow("detail_2",{field111:{value:"初始值"}});
//明细2添加一行并给新添加的行字段 field111赋值

//如果初始值类型是浏览框，则需要对应为浏览框的赋值格式：
{
  field110:{value:[ {'id':"id1",'name':"11"} ]},
  field112: {value:[ {'id':"id2",'name':"22"}]},
  ...
}
ModeForm.addDetailRow("detail_2",{field110:{value:[ {'id':"id1",'name':"11"} ]}});
```

## 参数说明

| 参数             | 参数类型   | 必须 | 说明  |
|----------------|--------|----|---|
| detailMark     | String | 是  | 明细表标示，明细1就是detail_1，以此递增类推  |
| initAddRowData | JSON   | 否  | 给新增后设置初始值，格式为{field110:{value:"11"},field112:{value:"22"},...}注意key不带下划线标示。 |

### 3.2 删除明细表指定行/全部行

```
delDetailRow: function(detailMark, rowIndexMark)
```

说明：此方法会清空明细已选情况，删除时没有提示“是否删除”的确认框

#### 参数说明

| 参数           | 参数类型   | 必须 | 说明                                |
|--------------|--------|----|-----------------------------------|
| detailMark   | String | 是  | 明细表标示，明细1就是detail_1，以此递增类推        |
| rowIndexMark | String | 是  | 需要删除的行标示，删除全部行:all,删除部分行: "0,1,2" |

样例

```
ModeForm.delDetailRow("detail_1", "all");           //删除明细1所有行
ModeForm.delDetailRow("detail_1", "3,6");           //删除明细1行标为3,6的行
```

### 3.3 选中明细指定行/全部行

```
checkDetailRow: function(detailMark, rowIndexMark,needClearBeforeChecked)
```

说明：此方法可灵活使用，依靠参数needClearBeforeChecked可实现清除选中的逻辑

#### 参数说明

| 参数                     | 参数类型    | 必须 | 说明                                  |
|------------------------|---------|----|-------------------------------------|
| detailMark             | String  | 是  | 明细表标示，明细1就是detail_1，以此递增类推          |
| rowIndexMark           | String  | 否  | 需要选中的行标示，选中全部行:all,选中部分行: "1, 2, 3" |
| needClearBeforeChecked | boolean | 否  | 是否需要清除已选                            |

样例

```
ModeForm.checkDetailRow("detail_2", "all"); //勾选明细2所有行
ModeForm.checkDetailRow("detail_2", "", true); //清除明细2所有已选
ModeForm.checkDetailRow("detail_2", "3,6", true); //清除明细2全部已选，再勾选行标为3,6的行
ModeForm.checkDetailRow("detail_2", "7", false); //保持已选记录，追加选中行标为7的行
```

### 3.4 获取明细行所有行标示

```
getDetailAllRowIndexStr: function(detailMark)
```

#### 参数说明

| 参数         | 参数类型   | 必须 | 说明                         |
|------------|--------|----|----------------------------|
| detailMark | String | 是  | 明细表标示，明细1就是detail_1，以此递增类推 |

样例

```
console.log(ModeForm.getDetailAllRowIndexStr("detail_2")); //输出1, 3...等等
```

### 3.5 获取明细已有行的数据库主键

```
getDetailRowKey: function(fieldMark)
```

此方法只对明细已有行生效，新增加的行/不存在的行返回-1

#### 参数说明

| 参数         | 参数类型   | 必须 | 说明   |
|------------|--------|----|--|
| detailMark | String | 是  | 字段标示，格式 field\${字段ID}_\${明细行号}，用于定位属于哪个明细表 |

样例

```
ModeForm.getDetailRowKey("field112_3"); //获取明细第四行主键
```

### 3.6 添加明细时默认复制最后一行记录

setDetailAddUseCopy: function(detailMark, needCopy)

说明：此方法在ready时调用，手动点添加时自动赋值最后行字段内容，覆盖节点前设置及默认值等，附件上传字段不予复制

注：E9是异步ready执行后点添加明细才生效，例如默认新增空明细无效；

后期标准产品直接支持明细复制，暂通过添加功能挂靠

#### 参数说明

| 参数         | 参数类型    | 必须 | 说明   |
|------------|---------|----|--|
| detailMark | String  | 是  | 字段标示，格式 field\${字段ID}_\${明细行号}，用于定位属于哪个明细表 |
| needCopy   | Boolean | 是  | 是否需要启用复制，true：启用，false：不启用                 |

#### 样例

```
jQuery(document).ready(function(){
    ModeForm.setDetailAddUseCopy("detail_1", true);
});
```

### 3.7 根据明细行标识获取序号(第几行)

最低版本要求: KB900190800

场景：根据明细行的下标，获取当前是第几行明细，可用于提示某某行异常等

getDetailRowSerailNum: function(mark, rowIndex)

#### 参数说明

| 参数       | 参数类型   | 必须 | 说明  |
|----------|--------|----|---|
| mark     | String | 是  | 明细表标示，支持两种格式 detail_ <i>dindex</i> 或者 <i>field</i> {fieldid}_\${rowIndex} |
| rowIndex | Int    | 是  | 行标识，第一种格式 detail_ <i>dindex</i> 才需要传此参数                                   |

```
ModeForm.getDetailRowSerailNum("detail_1", 3); //获取明细1下标为3的行序号
```

`ModeForm.getDetailRowSerailNum("field222_3");` //获取字段222对应明细表下标为3的行序号

注意： 不存在则返回-1

3.8 获取明细总行数

最低版本要求: KB900190800

`getDetailRowCount: function(detailMark)`

参数说明

| 参数         | 参数类型   | 必须 | 说明                           |
|------------|--------|----|------------------------------|
| detailMark | String | 是  | 明细表标示, 明细1就是detail_1, 以此递增类推 |

`ModeForm.getDetailRowCount("detail_1");`  
//输出明细总行数, 注意此结果只代表明细总行数, 不能 用作循环行

3.9 控制明细数据行的显示及隐藏

最低版本要求: KB900190800

注: 只是界面效果隐藏, 序号不会变化, 即被隐藏行的前后行序号会断层不连续

`controlDetailRowDisplay: function(detailMark, rowIndexMark, needHide)`

参数说明

| 参数           | 参数类型    | 必须 | 说明                                   |
|--------------|---------|----|--------------------------------------|
| detailMark   | String  | 是  | 明细表标示, 明细1就是detail_1, 以此递增类推         |
| rowIndexMark | String  | 是  | 需要选中的行标示, 选中全部行:all,选中部分行: "1, 2, 3" |
| needHide     | boolean | 是  | 是否隐藏行, true:隐藏, false:显示             |

`ModeForm.controlDetailRowDisplay("detail_1", "3,5", true);` //明细1行标为3,5的隐藏不显示

`ModeForm.controlDetailRowDisplay("detail_1", "all", false);` //明细1所有行不隐藏都显示

3.10 控制明细行check框是否禁用勾选



最低版本要求: KB900190800

注：后台配置的置灰行(不允许删除情况)，不支持通过此API控制

controlDetailRowDisableCheck: function(detailMark, rowIndexMark, disableCheck)

参数说明

| 参数           | 参数类型    | 必须 | 说明                                 |
|--------------|---------|----|------------------------------------|
| detailMark   | String  | 是  | 明细表标示，明细1就是detail_1，以此递增类推         |
| rowIndexMark | String  | 是  | 需要选中的行标示，选中全部行:all,选中部分行："1， 2， 3" |
| disableCheck | boolean | 是  | 是否禁用勾选，true:置灰禁止勾选， false:允许勾选     |

```
ModeForm.controlDetailRowDisableCheck("detail_1", "all", true);    //明细1
所有行check框置灰禁止选中

ModeForm.controlDetailRowDisableCheck("detail_1", "1,2", false);
//明细1行标为1,2的行释放置灰，允许勾选
```

3.11 获取明细选中行主键ID

最低版本要求: KB900190800

注意：新添加的行返回 空

getDetailCheckedRowKey: function(detailMark)

参数说明

| 参数         | 参数类型   | 必须   | 说明                         |
|------------|--------|------|----------------------------|
| detailMark | String | true | 明细表标示，明细1就是detail_1，以此递增类推 |

```
ModeForm.getDetailCheckedRowKey("detail_1"); //输出选中行101， 102...等等
```

3.12 获取明细选中行下标

最低版本要求: KB900190800

getDetailCheckedRowIndexStr: function(detailMark)

### 参数说明

| 参数         | 参数类型   | 必须 | 说明                         |
|------------|--------|----|----------------------------|
| detailMark | String | 是  | 明细表标示，明细1就是detail_1，以此递增类推 |

```
ModeForm.getDetailCheckedRowIndexStr("detail_1"); //输出选中行1, 3...等等
```

## 4. 浏览按钮类型相关接口

### 4.1 扩展浏览按钮取数接口参数值

最低版本要求: KB900190700

场景：控制浏览按钮可选数据范围，限定范围、依赖表单字段过滤数据范围等；对联想输入范围及弹窗选择 范围都生效；

实现方式：接口扩充的参数会通过url参数提交到服务端接口，需结合修改浏览按钮接口类方可生效

```
appendBrowserDataUrlParam: function(fieldMark, jsonParam)
```

### 参数说明

| 参数        | 参数类型   | 必须 | 说明                             |
|-----------|--------|----|--------------------------------|
| fieldMark | String | 是  | 字段标示，格式 field\${字段ID}_\${明细行号} |
| jsonParam | JSON   | 否  | 扩展的url参数，JSON中key-value格式      |

### 样例

```
ModeForm.appendBrowserDataUrlParam("field395",{"customerid":"2"});  
//给浏览按钮395请求后台数据时 增加url参数customerid
```

### 4.2 获取浏览按钮的显示值

获取浏览按钮的显示名称，多个则以splitChar字符分隔拼接成串；

```
getBrowserShowName:function(fieldMark,splitChar)
```

### 参数说明

| 参数        | 参数类型   | 必须 | 说明                             |
|-----------|--------|----|--------------------------------|
| fieldMark | String | 是  | 字段标示，格式 field\${字段ID}_\${明细行号} |
| splitChar | String | 否  | 分隔符，默认以逗号分隔                    |

样例

```
ModeForm.getBrowserShowName("field110"); //以逗号分隔获取浏览按钮字段显示值
```

## 5. 选择框类型相关接口

### 5.1 移除选择框字段选项

```
removeSelectOption: function(fieldMark, optionKeys)
```

#### 参数说明

| 参数         | 参数类型   | 必须 | 说明                             |
|------------|--------|----|--------------------------------|
| fieldMark  | String | 是  | 字段标示，格式 field\${字段ID}_\${明细行号} |
| optionKeys | String | 是  | 需要移除的Option选项key值，多个以逗号分隔      |

样例

```
ModeForm.removeSelectOption("field112", "3,4"); //移除选择框中id值为3/4的选项
//当optionKeys = "" 时，显示全部的下拉选项
```

### 5.2 控制选择框字段选项

```
controlSelectOption: function(fieldMark, optionKeys)
```

#### 参数说明

| 参数         | 参数类型   | 必须 | 说明                             |
|------------|--------|----|--------------------------------|
| fieldMark  | String | 是  | 字段标示，格式 field\${字段ID}_\${明细行号} |
| optionKeys | String | 是  | 完全控制选择框的选项范围                   |

样例

```
ModeForm.controlSelectOption("field112", "1,2,4"); //控制选择框只显示1/2/4
的选项
ModeForm.controlSelectOption("field112", ""); //清除选择框所有选项
```

5.3 获取选择框字段的显示值

最低版本要求: KB900190800

限定条件: 仅适用选择框类型字段

获取选择框类型的显示名称, 多个则以splitChar字符分隔拼接成串

```
getSelectShowName:function(fieldMark,splitChar)
```

参数说明

| 参数        | 参数类型   | 必须 | 说明                          |
|-----------|--------|----|-----------------------------|
| fieldMark | String | 是  | 字段标示, 格式 field{字段ID}_{明细行号} |
| splitChar | String | 否  | 分隔符, 默认以逗号分隔(只有复选框多选才会用到)   |

```
ModeForm.getSelectShowName("field10_0"); //获取选择框字段显示值
```

6.其它接口

6.1 表单顶部按钮、右键菜单置灰

说明: 设置表单顶部按钮、右键菜单置灰不可操作和恢复操作功能参数isDisabled

```
controlBtnDisabled: function(isDisabled)
```

参数说明

| 参数         | 参数类型    | 必须 | 说明                                |
|------------|---------|----|-----------------------------------|
| isDisabled | boolean | 是  | true: 按钮全部置灰不可操作,false: 恢复按钮可操作状态 |

```
function subimtForm(params){
  ModeForm.controlBtnDisabled(true);
  ...
  ModeForm.controlBtnDisabled(false);
}
```

## 6.2 根据字段ID获取字段信息

说明：根据字段ID获取字段信息，JSON格式，包括名称、类型、只读必填属性等

```
getFieldInfo:function(fieldid)
```

### 参数说明

| 参数        | 参数类型   | 必须 | 说明                    |
|-----------|--------|----|-----------------------|
| fieldMark | String | 是  | 字段ID，格式 field\${字段ID} |

返回值字段信息JSON重要属性说明

| 参数            | 说明                        |
|---------------|---------------------------|
| fieldhtmltype | 字段大类型(文本/多行文本...)         |
| fieldtype     | 1文本 2整数(11) 4金额转换 3浮点(38) |
| fieldname     | 字段数据库名称                   |
| fieldlabel    | 字段显示名                     |
| viewattr      | 字段属性(1:只读; 2: 可编辑; 3: 必填) |

```
ModeForm.getFieldInfo("field15612");
```

## 6.3 获取字段当前的只读/必填属性

此方法为实时获取字段显示属性，不是仅仅获取后台配置的字段属性；

如只想获取后台配置的字段属性，调用接口6.2取返回值viewattr属性

```
getFieldCurViewAttr:function(fieldMark)
```

### 参数说明

| 参数        | 参数类型   | 必须 | 说明                             |
|-----------|--------|----|--------------------------------|
| fieldMark | String | 是  | 字段标示，格式 field\${字段ID}_\${明细行号} |

```
ModeForm.getFieldCurViewAttr("field110_2");  
//获取明细字段属性，1：只读、2：可编辑、3：必填；
```

## 6.4 获取卡片的url参数

此方法实时获取顶部url里面的参数及卡片内部的一些参数；

```
getCardUrlInfo:function()
```

返回值字段信息JSON重要属性说明

| 参数             | 说明                                     |
|----------------|--|
| billid         | 数据id                                   |
| formId         | 表单id                                   |
| modelId        | 模块id                                   |
| modelName      | 模块名称                                   |
| modedatastatus | 草稿状态 0 非草稿 1 草稿                        |
| selectCategory | 文档上传目录                                 |
| type           | 模板类型 0显示模板 1 新建模板 2 编辑模板 3 监控模板 4 打印模板 |

```
ModeForm.getCardUrlInfo();
```

## 6.5 刷新卡片当前页面

此方法用于仅刷新当前卡片页面

```
reloadCard:function(params)
```

### 参数说明

| 参数     | 参数类型   | 必须 | 说明               |
|--------|--------|----|------------------|
| params | Object | 否  | 对象：{} 可以传一些新的键值对 |

```
ModeForm.reloadCard();
```

## 6.6 外部调用卡片保存方法

注意：此方法主要用于页面扩展按钮

此方法支持外部调用,增加了两个控制参数，1、控制页面保存后是否跳转显示页面；2、支持保存后函数回调，返回billid

```
doCardSubmit:function(pageexpandid, issystemflag, btntype,isRefreshto,callback)
```

## 参数说明

| 参数           | 参数类型     | 必须 | 说明  |
|--------------|----------|----|---|
| pageexpandid | string   | 是  | 页面扩展的id(传哪个扩展id 就是执行那个按钮配置的接口)              |
| issystemflag | string   | 否  | 是否是调用系统默认的保存扩展按钮保存, 0表示不调用, 1表示调用。其他也是视为不调用 |
| btntype      | string   | 否  | 按钮类型, 默认为空, 建议直接传空值占位                       |
| isRefreshto  | boolean  | 否  | 保存后是否跳转到显示页面, 默认为true, 跳转到显示页面              |
| callback     | function | 否  | 保存完成后回调函数, 返回billid, 然后自己写自己的逻辑             |

```
ModeForm.doCardSubmit('14169','0','',false,function(billid){
    console.log("===billid===",billid)
});
//保存方法, 保存后不跳转显示页面, 打印出billid
```

## 6.7 可控制显示时间的message信息

```
showMessage: function(msg, type, duration)
```

## 参数说明

| 参数       | 参数类型   | 必须    | 说明  |
|----------|--------|-------|---|
| msg      | String | true  | 提示信息内容  |
| type     | int    | false | 提示类型, 1(警告)、2(错误)、3(成功)、4(一般), 默认为1, 不同类型提示信息效果不同 |
| duration | Float  | false | 多长时间自动消失, 单位秒, 默认为1.5秒                            |

```
ModeForm.showMessage("结束时间需大于开始时间"); //警告信息, 1.5s后自动消失
```

```
ModeForm.showMessage("运算错误", 2, 10); //错误信息, 10s后消失
```

## 6.8 系统样式的Confirm确认框

```
showConfirm: function(content, okEvent, cancelEvent, otherInfo={})
```

### 参数说明

| 参数          | 参数类型     | 必须 | 说明          |
|-------------|----------|----|-------------|
| content     | String   | 是  | 确认信息        |
| okEvent     | Function | 是  | 点击确认事件      |
| cancelEvent | Function | 否  | 点击取消事件      |
| otherInfo   | Object   | 否  | 自定义信息(按钮名称) |

```
ModeForm.showConfirm("确认删除吗?", function(){
    alert("删除成功");
});

ModeForm.showConfirm("请问你是否需要技术协助?", function(){
    alert("点击确认调用的事件");
}, function(){
    alert("点击取消调用的事件");
},{
    title:"信息确认",           //弹确认框的title, 仅PC端有效
    okText:"需要",              //自定义确认按钮名称
    cancelText:"不需要"        //自定义取消按钮名称
})
```

### 6.9 系统样式的Modal弹出框

最低版本要求: KB900190800

```
showModalMsg:function(title,msg,type)
```

### 参数说明

| 参数    | 参数类型        | 必须    | 说明  |
|-------|-------------|-------|---|
| title | String      | true  | 提示信息头部  |
| msg   | String、html | true  | 提示信息内容, 支持html格式                                  |
| type  | int         | false | 提示类型, 1(一般)、2(错误)、3(成功)、4(警告), 默认为1, 不同类型提示信息效果不同 |



```
ModeForm.showModalMsg("系统提示","提示的信息内容",3) //成功提示信息
```

```
ModeForm.showModalMsg("系统提示","<div style='color:red;'>提示的信息内容</div>",2) //错误提示信息
```

## 7.历史E8代码块的相关兼容

以下方法不推荐使用，只为兼容历史代码块，上述ModeForm中API都涵盖包括

建议全部改造通过ModeForm接口实现，操作DOM容易引发问题及后续版本升级可能不兼容

### 7.1 提交事件执行自定义函数

比如checkCustomize 会继续执行,根据函数返回值判断是否阻塞提交，返回值true：继续流转，false:阻断提交

开发示例

```
window.checkCustomize =()=>{
    var flag = true;    //
    ...
    return flag;
}
//请使用普通定义方法
function checkCustomize (){
    var flag = true;    //
    ...
    return flag;
}
```

### 7.2 字段值变化触发事件bindPropertyChange

说明：与E8一致，依赖DOM，字段值变化时触发此函数; 自定义函数默认传递以下三个参数，参数1：触发字段的DOM对象，参数2：触发字段的标示(field27563等)，参数3：修改后的值

样例

```
jQuery(document).ready(function(){
    jQuery("#field27563").bindPropertyChange(function(obj,id,value){
        console.log("tri...",obj,id,value);
    });
});
```

```
});
```

### 7.3 明细新增行渲染后触发事件

最低版本要求: KB900190700

说明：重载`_customAddFun"+groupid+"函数`，`groupid`从1开始递增，1代表明细1；不论手动添加、联动添加、接口添加都会触发

注意：此方法内是可以获取到新增行的DOM结构，自定义函数默认传参`addIndexStr`为新增行的标示串；如联动一次性增加多行，此方法只触发一次，参数`addIndexStr`值为多行的行标示，逗号分隔；

样例

```
function _customAddFun1(addIndexStr){  
    // 明细1新增成功后触发事件，addIndexStr即刚新增的行标示， 添加多行为(1,2,3)  
    console.log("新增的行标示: "+addIndexStr);  
}
```

### 7.4 明细删除行渲染后触发事件

最低版本要求: KB900190700

说明：重载`_customDelFun"+groupid+"函数`，`groupid`从1开始递增，1代表明细1；不论手动删除、接口删除 都会触发

样例

```
function _customDelFun1(){ // 明细1删除成功后触发事件  
    console.log("删除明细");  
}
```

### 7.5 jQuery操作

不推荐使用,历史jQuery("#field111").val()取值、赋值等操作已兼容

### 7.6 JS原生操作

不允许，会引发不可控各种异常

类似`document.write("");document.getElementById("field111").value`操作需要调整为`ModeForm`接口操作

## 8 值变更触发事件接口

### 8.1 字段值变化触发事件

字段值变化即会触发所绑定的函数，可多次绑定

bindFieldChangeEvent: function(fieldMarkStr,funobj)

#### 参数说明

| 参数           | 参数类型     | 必须 | 说明   |
|--------------|----------|----|--|
| fieldMarkStr | String   | 是  | 绑定字段标示，可多个拼接逗号隔开，例如：field110(主字段),field111_2(明细字段).....                      |
| funobj       | Function | 是  | 字段值变化触发的自定义函数，函数默认传递以下三个参数，参数1：触发字段的DOM对象，参数2：触发字段的标示(field27555等)，参数3：修改后的值 |

样例

```
ModeForm.bindFieldChangeEvent("field27555,field27556",function(obj,id,value){
    console.log("ModeForm.bindFieldChangeEvent--",obj,id,value);
});
```

### 8.2 明细字段值变化触发事件

绑定后对新添加的明细行字段以及加载的已有行明细字段，值变更触发所绑定的事件

bindDetailFieldChangeEvent: function(fieldMarkStr,funobj)

#### 参数说明

| 参数           | 参数类型     | 必须 | 说明   |
|--------------|----------|----|--|
| fieldMarkStr | String   | 是  | 绑定的明细字段标示，不能有下列线标示，可多个拼接逗号隔开，例如：field110,field111                |
| funobj       | Function | 是  | 字段值变更触发自定义函数，函数默认传递以下三个参数，参数1：字段标示(field27583)，参数2：行标示，参数3：修改后的值 |

样例

```
jQuery(document).ready(function(){
    ModeForm.bindDetailFieldChangeEvent("field27583,field27584",
        function(id,rowIndex,value){
            console.log("ModeForm.bindDetailFieldChangeEvent--",id,rowInd
ex,value);
        });
});
```

9. 关闭打开的窗口

9.1 窗口打开的两种方式

- 弹出窗口打开
- 滑动窗口打开

```
//关闭弹出窗口打开
window.close();
//关闭滑动窗口打开
ModeForm.closePageBySlide();
```

10、新增的一部分方法

10.1 获取当前用户信息

```
最低版本要求: KB900190800

getCurrentUserInfo: function()
```

返回值字段信息JSON重要属性说明

| 参数             | 说明   |
|----------------|------|
| userid         | 用户id |
| username       | 用户名  |
| jobs           | 岗位名称 |
| subcompanyid   | 分部id |
| subcompanyname | 分部名称 |

|          |      |
|----------|------|
| deptid   | 部门id |
| deptname | 部门名称 |
| icon     | 人员头像 |

样例

```
ModeForm.getCurrentUserInfo(); //获取用户信息
```

10.2 文本字段可编辑状态，当值为空显示默认灰色提示信息，鼠标点击输入时 提示消失

最低版本要求: KB900190800

限定条件：仅支持单行文本、整数、浮点数、千分位、多行文本字段(非html)字段类型；支持主字段及明细字 段

```
setTextFieldEmptyShowContent:function(fieldMark,showContent)
```

参数说明

| 参数          | 参数类型   | 必须 | 说明                         |
|-------------|--------|----|----------------------------|
| fieldMark   | String | 是  | 字段标示，格式 field{字段ID}_{明细行号} |
| showContent | String | 是  | 空值时显示的提示信息，灰色              |

```
jQuery(document).ready(function(){
    ModeForm.setTextFieldEmptyShowContent("field27555", "单文本默认提示信息1");
    ModeForm.setTextFieldEmptyShowContent("field27566", "多文本默认提示2");

    ModeForm.setTextFieldEmptyShowContent('field23824_0','明细表提示信息')
});
```

10.3 控制Radio框字段打印是否仅显示选中项文字

最低版本要求: KB900190800

仅支持选择框中单选框类型，打印场景，是否仅显示选中项文字，都未选中显示空

```
controlRadioPrintText: function(fieldMark)
```

参数说明

| 参数        | 参数类型   | 必须 | 说明                                 |
|-----------|--------|----|------------------------------------|
| fieldMark | String | 是  | 字段标示, 格式 field\${字段ID} ,支持明细表字段整列, |

`ModeForm.controlRadioPrintText("field27555");` //单选框字段field27555打印只  
显示文字



```
<!-- -->  
<!--ModeForm.controlCardInnerTabSelect("68976"); -->  
<!-- -->
```

## 11、注册自定义事件

### 11.1 注册拦截事件，指定动作执行前触发，并可阻断/放行后续操作

最低版本要求: KB900190800

支持多次注册，按注册顺序依次执行；支持异步ajax，避免请求卡住

场景1：明细添加行、删除行前，执行自定义逻辑并阻断/允许后续操作

| 动作类型                 | 说明            |
|----------------------|---------------|
| ModeForm.OPER_ADDROW | 添加明细行，需拼明细表序号 |
| ModeForm.OPER_DELROW | 删除明细行，需拼明细表序号 |

接口名称及参数说明

`registerCheckEvent:function(type,fun)`

| 参<br>数 | 参数类<br>型 | 必<br>须 | 说明   |
|--------|----------|--------|--|
| type   | String   | 是      | 动作类型(详见上表)，多个逗号分隔  |
| fun    | Function | 是      | 自定义函数，此函数入参为callback，执行自定义逻辑完成或异步ajax的<br>success 函数体内，放行需调用callback，不调用代表阻断后续操作 |

样例

```
jQuery().ready(function(){
    ModeForm.registerCheckEvent(ModeForm.OPER_ADDROW+"1", function(callback){
        alert("添加明细1前执行逻辑，明细1则是OPER_ADDROW+1，依次类推")
        callback(); //允许继续添加行调用callback，不调用代表阻断添加
    });
    ModeForm.registerCheckEvent(ModeForm.OPER_DELROW+"2", function(callback){
        alert("删除明细2前执行逻辑")
        callback(); //允许继续删除行调用callback，不调用代表阻断删除
    });
})
```

11.2 注册钩子事件，指定动作完成后触发

最低版本要求: KB900190800

支持多次调用注册，按注册的先后顺序依次执行

| 动作类型                   | 说明            |
|------------------------|---------------|
| ModeForm.ACTION_ADDROW | 添加明细行，需拼明细表序号 |
| ModeForm.ACTION_DELROW | 删除明细行，需拼明细表序号 |

接口名称及参数说明

registerAction: function(actionname, fn)

| 参数         | 参数类型     | 必须 | 说明        |
|------------|----------|----|-----------|
| actionname | String   | 是  | 动作类型，详见上表 |
| fn         | Function | 是  | 触发事件      |

```
...
ModeForm.registerAction(ModeForm.ACTION_ADDROW+"1", function(index){
    alert("添加行下标是"+index);
}); //下标从1开始，明细1添加行触发事件，注册函数入参为新添加行下标

ModeForm.registerAction(ModeForm.ACTION_DELROW+"2", function(arg){
    alert("删除行下标集合是"+arg.join(","));
}); //下标从1开始，明细2删除行触发事件
...
```

### 12.1 字段区域绑定动作事件

最低版本要求: KB900190800

推荐使用值变化事件实现开发,因为此接口点击、双击等动作不是绑定到字段元素，是字段所在单元格区域即会 触发

| 类型         | 说明                     |
|------------|------------------------|
| onblur     | 失去焦点事件，仅支持单行文本类型       |
| onfocus    | 获取焦点事件，仅支持单行文本字段类型     |
| onclick    | 单击事件，字段所在单元格区域单击触发     |
| ondblclick | 双击事件，字段所在单元格区域双击触发     |
| mouseover  | 鼠标移入事件，鼠标移入字段所在单元格区域触发 |
| mouseout   | 鼠标移出事件，鼠标移出字段所在单元格区域触发 |

#### 参数说明

| 参数       | 参数类型     | 必须 | 说明                                   |
|----------|----------|----|--------------------------------------|
| type     | String   | 是  | 动作类型，见上表                             |
| fieldids | String   | 是  | 字段id集合，多个逗号分隔，明细字段不加下划线对所有行生效        |
| fn       | Function | 是  | 触发函数，此函数入参接收两个参数，fieldid以及rowIndex行号 |

#### 样例

```
ModeForm.bindFieldAction("onfocus", "field111,field222", function(fieldid, rowIndex){
    console.log("单行文本字段111获取焦点触发事件");
    console.log("明细第"+rowIndex+"行字段fieldid222获取焦点触发事件");
});

ModeForm.bindFieldAction("onclick", "field333", function(){
    console.log("浏览按钮字段单击触发事件，不是指点放大镜选择，是整个字段所在单元格区域单击都会触发");
});
```

### 12.2 自定义代理渲染单行文本框字段



最低版本要求: KB900190800

此接口仅对单行文本框字段类型生效，即数据库字段类型为varchar

显示效果、事件、字段值交互都可自行控制，通过接口changeFieldValue修改的可编辑字段值也会正常入库

此接口传入的组件，产品会传入此字段依赖的相关props，具体可通过React Developer Tools自行抓取，按需调用

proxyFieldComp: function(fieldMark, el, range)

参数说明

| 参数        | 参数类型       | 必须 | 说明                                  |
|-----------|------------|----|-------------------------------------|
| fieldMark | String     | 是  | 字段标示，格式 field{字段ID}_{明细行号}          |
| el        | React Comp | 是  | 渲染的组件                               |
| range     | String     | 否  | 作用范围，默认全部，(1:只读、2:可编辑、3:必填),组合后逗号分隔 |

样例

```
ModeForm.proxyFieldComp("field111", React.createElement("div",{
  style:{background:"red"},
  children:"子内容"
})); //字段111在只读、可编辑、必填情况下自定义渲染

ModeForm.proxyFieldComp("field222_1", "<div>自定义渲染字段</div>", "2,3");
// 明细某行字段222再可编辑、必填情况下自定义渲染
```

12.3 自定义追加渲染表单字段

最低版本要求: KB900190800

在标准字段展现内容的基础上，after方式追加渲染自定义组件

此接口参数说明与用法，与接口proxyFieldComp类同

afterFieldComp: function(fieldMark, el, range)

样例

```
ModeForm.afterFieldComp("field111", React.createElement("a",{
```

```
href: "/test.jsp?userid="+ModeForm.getFieldValue("field222"),
children: "自定义链接"
});
// 字段111在只读、可编辑、必填情况下,追加渲染个自定义链接,链接参数依赖表单其它字段值
```

### 12.4 控制日期浏览按钮的可选日期范围

最低版本要求: KB900190800

仅支持日期类型,控制手动选择时的可选日期范围

controlDateRange: function(fieldMark,start,end)

#### 参数说明

| 参数        | 参数类型   | 必须 | 说明  |
|-----------|--------|----|---|
| fieldMark | String | 是  | 字段标示, 格式 field{字段ID}_{明细行号}                             |
| start     | String | 是  | 支持两种格式, 第一种标准的日期格式, 比如2019-05-28, 第二种整数, 相比于当前日期往前/后多少天 |
| end       | String | 否  | 格式与start参数一致  |

#### 样例

```
ModeForm.controlDateRange("field111", -5, 10); // 限定日期可选范围, 往前5天, 往后10天

ModeForm.controlDateRange("field111", 0, '2019-12-31'); // 限定今天至本年

ModeForm.controlDateRange("field222_0", '2019-05-01', '2019-05-31'); // 明细字段, 限定当月
```

### 12.5 复写浏览按钮组件的props

最低版本要求: KB900190800

仅支持浏览按钮类型, 谨慎使用, 完全重写覆盖浏览按钮的props

只是传递props, 具体传递的属性实现何种需求由组件内部控制

overrideBrowserProp:function(fieldMark,jsonParam)

#### 参数说明

| 参数        | 参数类型   | 必须 | 说明                          |
|-----------|--------|----|-----------------------------|
| fieldMark | String | 是  | 字段标示, 格式 field{字段ID}_{明细行号} |
| jsonParam | JSON   | 是  | 扩展或复写的props参数               |

```
ModeForm.overrideBrowserProp("field111",{
    onBeforeFocusCheck: function(success, fail){/**/}

});    //复写浏览按钮字段111的props
```

## 13、全局的一些方法介绍

注：全局方法e9项目下可以直接调用

### 13.1 页面loading

- 1、`ecCom.WeaLoadingGlobal.start()` ; //开始加载 页面出现加载效果
- 2、`ecCom.WeaLoadingGlobal.end()`; //结束加载, 但是没去掉遮罩
- 3、`ecCom.WeaLoadingGlobal.destroy()`; //销毁加载, 去掉了遮罩  
//具体使用根据业务场景需求, 正常使用可以直接调用1、3就可以

### 13.2 e9公共异步请求方法

参数说明

| 参数     | 说明   | 类型     | 可选值                  | 默认     |
|--------|------|--------|----------------------|--------|
| url    | 接口路径 | string |                      |        |
| method | 请求类型 | string | 'GET'                | 'POST' |
| params | 请求参数 | object | { }                  |        |
| type   | 响应类型 | string | 'json' 'text' 'json' |        |

```
ecCom.WeaTools.callApi('/api/ec/dev/table/datas', 'POST', params).then(fu  
nction(data) {  
    //请求之后逻辑  
    console.log(data) // jsonObj or text  
});
```

# 以下查询列表的API文档

## e9查询列表方法说明

注：查询列表的全局对象为【ModeList】

操作后，请点击右键保存数据！

在调用该代码块方法时，请使用格式： `javascript: fun();`

在【查询】->【字段定义的自定义链接地址】中使用时，链接路径为【自定义】，链接路径格式： `javascript: fun();`

在【模块】->【页面扩展】中使用时，选择打开方式为【其他】，链接目标来源为【手动输入】，链接目标地址格式： `javascript: fun();`

### 1、清除选中的checkbox

`clearChecked: function()`

描述: 清除选中的checkbox

样例：

```
ModeList.clearChecked();
```

### 2、选中的checkbox的值

`getCheckedID: function()`

描述: 选中的checkbox的值（主表数据ID）

样例：

```
var ids = ModeList.getCheckedID();  
//多个数据ID以英文半角逗号分开；例如：`1,2,3,4`
```

### 3、选中的checkbox的包含明细的值

`getCheckedIDWithDetail: function()`

描述: 选中的checkbox的带有明细的值（主表数据ID\_明细数据ID）

样例：

```
var ids = ModelList.getCheckedIDWithDetail();
```

返回值说明

*//多个数据ID以英文半角逗号分开；如果主表没有明细格式如：主表数据ID + “\_”；*

*//例如：`1\_1,1\_2,2\_1` 或者 `1\_`*

#### 4、获取查询列表Id

```
getCustomID: function()
```

描述: 获取查询列表Id

样例:

```
var customId = ModelList.getCustomID();
```

返回值说明

*// 获取查询列表Id*

*//例如：`163`*

#### 5、获取查询列表和模块、表单Id

```
getCustomIdWithModeIDAndFormID: function()
```

描述: 获取查询列表Id+模块Id+表单Id

样例:

```
var customId = ModelList.getCustomIdWithModeIDAndFormID();
```

返回值说明

*//查询列表Id+模块Id+表单Id*

*//例如：`{customId:customId,formId:formId,modeId:modeId}`*

#### 6、获取表单ID

```
getFormID: function()
```

描述: 获取查询列表对应表单ID

样例:

```
var formId = Modelist.getFormID();
```

返回值说明

// 获取查询列表对应表单ID

// 例如: `-163`

## 7、获取模块ID

getModelID: function()

描述: 获取查询列表对应模块ID

样例:

```
var modeId = Modelist.getModelID();
```

返回值说明d

// 获取查询列表对应模块ID

// 例如: `66`

## 8、获取当页表格json数据

getTableDatas: function()

描述: 获取当页表格json数据, 不包含生成的span数据

样例:

```
var datas = Modelist.getTableDatas();
```

返回值说明

// 表格的json 数据数组

// 例如:

```
[{"bm": "777", "dwb": "ddd", "id": "8"}]
```

## 9、获取当页表格json数据包含span

getTableDatasWithSpan: function()

描述: 获取当页表格json数据，包含生成的span数据

样例:

```
var datas = Modelist.getTableDatasWithSpan();
```

返回值说明

// 表格的json 数据数组

// 例如:

```
[{"bm": "777", "bmspan": "<a href=\"\">项目部</a>",  
  "id": "8", "idspan": "8",}]
```

## 10、未选中的checkbox的值

getUnCheckedID: function()

描述: 未选中的checkbox的值（主表数据ID）

样例:

```
var ids = Modelist.getUnCheckedID();
```

返回值说明

// 多个数据ID以英文半角逗号分开； 例如: `1,2,3,4`

## 11、未选中的checkbox包含明细值

getUnCheckedIDWithDetail: function()

描述: 未选中的checkbox的带有明细的值（主表数据ID\_明细数据ID）

样例:

```
var ids = Modelist.getUnCheckedIDWithDetail();
```

## 12、刷新表格

reloadTable: function()

描述: 重新加载表格

样例:

```
ModeList.reloadTable();
```

## 13、全选

setAllChecked: function()

描述: 全选

样例:

```
ModeList.setAllChecked();
```

返回值说明

// 多个数据ID以英文半角逗号分开; 如果没有明细格式: 主表数据ID + “\_”

// 例如: `1\_1,1\_2,2\_1` 或者 `1\_`

## 14、可控制显示时间的message信息

showMessage: function(msg, type, duration)

### 参数说明

| 参数       | 参数类型   | 必须    | 说明  |
|----------|--------|-------|---|
| msg      | String | true  | 提示信息内容  |
| type     | int    | false | 提示类型, 1(警告)、2(错误)、3(成功)、4(一般), 默认为1, 不同类型提示信息效果不同 |
| duration | Float  | false | 多长时间自动消失, 单位秒, 默认为1.5秒                            |

```
ModeList.showMessage("结束时间需大于开始时间"); //警告信息, 1.5s后自动消失
```



```
Modelist.showMessage("运算错误", 2, 10); //错误信息, 10s后消失
```

## 15、系统样式的Confirm确认框

```
showConfirm: function(content, okEvent, cancelEvent, otherInfo={})
```

### 参数说明

| 参数          | 参数类型     | 必须 | 说明          |
|-------------|----------|----|-------------|
| content     | String   | 是  | 确认信息        |
| okEvent     | Function | 是  | 点击确认事件      |
| cancelEvent | Function | 否  | 点击取消事件      |
| otherInfo   | Object   | 否  | 自定义信息(按钮名称) |

```
Modelist.showConfirm("确认删除吗?", function(){
    alert("删除成功");
});

Modelist.showConfirm("请问你是否需要技术协助?", function(){
    alert("点击确认调用的事件");
}, function(){
    alert("点击取消调用的事件");
},{
    title:"信息确认",           //弹确认框的title, 仅PC端有效
    okText:"需要",              //自定义确认按钮名称
    cancelText:"不需要"        //自定义取消按钮名称
})
```

## 16、打开自定义对话框

此方法用来打开一个自定义对话框;

```
openCustomDialog:function(prop,buttons?)
```

//如果是内部iframe里定义的方法,直接写方法名,外部添加代码块方式添加的方法,需要在方法名前面加上base.

```
let buttons=[{btnname:'保存',callfun:'saveDialog'},
              {btnname:'新建',callfun:'base.add'},
              {btnname:'关闭',callfun:'closeDialog'}];
let style={width:300,height:600};
let prop={title:'测试jsp',url:'/formmode/test.jsp',style:style};
```

```
ModeForm.openCustomDialog(prop,buttons)
```

返回值字段信息JSON重要属性说明

| 参数      | 说明  |
|---------|---|
| prop    | Object对象 参考WeaDialog对应的参数说明 title和url必传,分别为对话框标题和内嵌的jsp页面 |
| buttons | [] 对话框的按钮及对应的方法名称   |

prop字段信息JSON重要属性说明(其他属性请参考WeaDialog对应的参数说明)

| 参数            | 说明                | 类型     | 是否必填 | 默认                       |
|---------------|-------------------|--------|------|--------------------------|
| title         | Dialog 标题         | string | 是    |                          |
| url           | 内嵌的jsp页面          | string | 是    |                          |
| style         | Dialog样式          | Object | 否    |                          |
| icon          | Dialog顶部标题的图标     | string | 否    | icon-coms-ModelingEngine |
| iconBgcolor   | Dialog顶部标题的图标的背景色 | string | 否    | 96358a                   |
| iconFontColor | Dialog顶部标题颜色      | string | 否    | fff                      |

## 17、关闭自定义对话框

此方法用来关闭一个自定义对话框;

```
closeCustomDialog:function()
```

```
//该方法在嵌入的外部页面中自行调用  
parent.ModeForm.closeCustomDialog();
```