# CoolGrid表格插件

## CoolGrid表格插件主要功能

实现字段后台排序，整表保存，CRUD，分页翻页功能，单元格控件自定义。

## Grid表格插件json参数定义

### 1.2.1所有参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数值 | 例子 | 备注 |
| url | 字符串 | url: './studentServlet' | 对应后台请求地址(目前就是后台servlet的url pattern) |
| sortOrder[optional] | ‘asc’/’desc’ | sortOrder:’asc’ | 指定排序的升降序，不指定默认为desc |
| activeSortCol[optional] | 列名字符串(数据库对应列名) | activeSortCol:’productID’ | 指定根据哪个字段排序，不指定默认为sortable为true的第一个字段 |
| width[optional] | 像素值 | width:700 | table的宽度 |
| pageSize | 数字 | pageSize:10 | 指定分页大小 |
| colModel | 复合型json | 详细见下文 | 列属性 |
| databaseTableName | 字符串 | databaseTableName:’test1’ | 表格对应的数据库表名 |
| queryParams | json数组 | queryParams:[{name:’id’,value:’123’},{…}] | 查询参数 |
| saveTableEnable[optional] | true | saveTableEnable:true | 是否添加保存全表按钮，默认不添加 |
| queryModel[optional] | json数组 | 详见下文 | 指定此参数会在表格上方显示一个查询框，默认没有 |
| detailModel[optional] | json数组或’get’ | detailModel:’get’  detailModel:[{}{}] | 只有当列中存在type=”detail”时生效，详见下文 |
| insertModel[optional] | json数组或者’get’ | insertModel:’get’;  insertModel:[{}{}] | 是否允许新增数据，当设置为’get’时，insertModel从后台获得，也可以在前台直接配置 |
| colResize[optional] | true | colResize:true | 如果为true，则该表格的列宽可调整 |

### 1.2.2 colModel参数详解

参数名：colModel；

参数值: **[ {type:** **‘’, display: ‘’, name: ‘’, width: ‘’, sortable: ‘’, align: ‘’, opHtml: ‘’,key:true },{…},{…} ]** ：

属性说明如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数值 | 例子 | 备注 |
| type | data|delete|detail | type=’data’;  type=’ delete|detail; | 单元格控件的类型，可以任意类型组合,按照界面显示的从左向右的顺序配置 |
| display | 字符串 | display:’商品名’ | 是table表头的显示列名 |
| name | 字符创 | name:’productName’ | 在数据库中对应的列名 |
| sortable[optional] | true | sortable:true | 该列是否支持排序，默认为false |
| align[optional] | center/left/right, | align:’center’ | 对齐方式,默认为center |
| opHtml[optional] | html代码 | opHtml:”<a href=”xxx.xx”>xxx</a>” | 只有当type指定为undefined时该属性作为undefined控件的html代码添加到该列中 |
| key[optional] | true | key:true | 指定该列为主键，不是主键的列没有此属性,可指定多列，既多主键的情况。 |
| width | 像素值 | width:40 | 列宽度，该值为百分比 |
| editable[optional] | false | editable:false | 该列是否可编辑，该属性只在saveTableEnable属性为true时生效，默认为true |

### 1.2.3 queryModel参数详解

参数名：queryModel

参数值：{ legend:’学生信息’,data:[{type:’data’,display:’学号’,name:’studentID’},{},{}…] }

属性说明如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数值 | 例子 | 备注 |
| legend | 字符创 | legend=’学生信息’ | 区域名字 |
| type | data|date|comboBox | type=’data’ | 该查询条件是什么类型的 |
| display | 字符串 | display:’学号’ | 查询字段显示什么名字 |
| name | 字符串 | name:’studenID’ | 查询的字段对应数据库列名 |

### 1.2.4 detailModel参数详解

参数名：detailModel；

参数值: **[ { display: ‘’, name: ‘’},{…},{…} ]** ：

属性说明如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数值 | 例子 | 备注 |
| type | text|bool|number | type=’number’ | 用于修改该属性的控件类型 |
| dispaly | 字符串 | display=’学号’ | 该属性的显示名称 |
| name | 字符串 | name=’ID’ | 数据库对应的属性名 |
| key[optional] | true | key=’true’ | 该属性是否为主键，若为主键则不可修改 |

### 1.2.5 insertModel参数详解

参数名： insertModel；

参数值: **[ {type:** **‘’, display: ‘’, name: ‘’, key:true },{…},{…} ]** ：

属性说明如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数值 | 例子 | 备注 |
| type | text|bool|number | type=’number’ | 用于修改该属性的控件类型 |
| dispaly | 字符串 | display=’学号’ | 该属性的显示名称 |
| name | 字符串 | name=’ID’ | 数据库对应的属性名 |
| nullable | NO | key=’NO’ | 该属性是否为可为空，一个model里至少有一个 |

## CoolGrid初始化示范

$("#divID").coolGrid({

url: './servlet',

colModel :

[

{type:'checkbox|view',display: '操作', name : 'operator', width : 40}

{type:'data',display: '申请人', name : 'applyPeople', width : 40},

{type:'operator',display: '任务号', name : 'taskNum', width : 40, sortale:true}

]

sortorder: 'asc',

activeSortCol:'taskNum'

title: '示范表格',

width: 760,

pageSize: 5

});

## 前台Ajax Post调用后台Servlet的传入参数

对于ColModel中么一个JSON对象，其Type属性为insert或delete或update时，调用其相应的插入删除和修改的Javascript函数。此时必然要调用后台的Servlet做相应的增删改操作，CoolGrid是用JQuery封装的$.ajax({});方法，以post形式提交数据，就删除操作而言，其函数形式可能如下：

$.ajax({

url:"../SelectServlet",//发送请求地址

data:$.toJSON({opParam:"delete",dataTable:"sfcMaterialOrder",queryParams:{…},pageParam:{…}}),

type:'post',//请求方式

dataType:'json',

contentType: 'application/json;charset=utf-8',

success:**function**(data){

//若成功获得数据则加载表格

alert("操作成功！已删除数据。");

},

error: **function**(xhr) {alert("Single select page got fucked up!");}

});

下面对于$.ajax({});所传入的参数格式和含义进行说明：

### 1.4.1 所有参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数值 | 例子 | 备注 |
| url | 字符串 | "/SelectServlet" | 后台所要调用的Servlet |
| data | 复合型JSON串 | $toJSON(…)详见下文 | 操作后台数据库的实际参数 |
| type | 'post' |  | 请求提交方式，强制为post提交 |
| datatype | 'json' |  | 预期服务器返回的数据类型，在插件中强制指定为JSON格式。 |
| contenttype | 'application/json;charset=utf-8' |  | 发送信息至服务器时内容编码类型，在插件中强制指定为JSON格式。 |
| success:function(){}; | / | alert("操作成功！已删除数据。"); | 异步请求成功后的回调函数 |
| error:function(xhr){}; | / | alert("error!"); |  |

### 1.4.2 data参数详解

参数名：data；

参数值：根据用户选择的是单纯的翻页浏览及初始化或者增删改操作，参数构成不尽相同。

**对于删除操作：**

{opParam:"delete",dataTable:"detail",queryParams:[{name:’…’,value:’…’},{name:’…’,value:’…’},…]}

opParam对应用户的页面操作参数，在后台获取后根据if判断页面操作参数后进入不同的函数块；dataTable对应该条请求所要操作的数据表；queryParams对应选择操作记录行时SQL语句where后面的参数,为一个JSON数组，由表单中serializeArray()方法序列化而来，其格式为:

[{'name':'…','value':'…'},{'name':'…','value':'…'},{'name':'…','value':'…'}]。

**对于修改操作：**

{opParam:"update",dataTable:" detail ",queryParams:[{},{}],changeParams:[{},{}]}

前三个参数皆和删除操作中的一样，而changeParams则为SQL语句中对应的Update TableName Set后面的参数，也由表单中serializeArray()方法序列化而来，格式和queryParams一样。

**对于全表保存操作：**

{opParam:"saveTable",dataTable:" detail ",queryParams:[{data:[{},{}]}],changeParams:[{data:[{},{}]}]}

全表保存的参数跟修改操作中的类似，只不过queryParams和changeParams编程了数组的数组，多了一层嵌套。两个Index相等的data就是修改操作中对应的queryParams和changeParams。

**对于插入操作：**

{opParam:"insert",dataTable:" detail ",dataParams:{}}

格式和前面所对应的同名参数意义和格式相同。

**而对于单纯的浏览翻页和页面初始化操作：**

{opParam:"view",dataTable:" detail ",queryCols:{[…]},

pageParam:{currentPage:1,pageSize:5,totalPage:2},queryParams:{},sortParams:{sortCol: "materialOrder",order:”asc”}}

queryCols是一个字符串数组，其含义为你所要查询的列名;pageParam对应于分页参数，currentPage为当页页数，pageSize为单页显示最大记录数，totalPage为记录的页面总数；sortParams对应排序参数,sortCol对应于你想进行排序的字段，order对应为升序还是降序。

对于增删改操作，一般都会在请求成功后的success回调函数中调用opParam参数为"view"的ajax请求。

另外，以上参数的顺序不影响后台参数的获取。

## 后台返回json格式定义

### 1.5.1 后台返回json格式

{ dataSet: [ {},{} ], pageParam : {currentPage:1, totalPage:2} }

### 1.5.2 后台返回json参数详解

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数值 | 例子 | 备注 |
| dataSet | json数组 | [{’ID’:’123’,’name’:’cloud’},{}] | 是后台返回数据表数据的主要载体 |
| pageParam | json | {currentPage:1,totalPage:2} | 翻页有关的参数 |