

## 说明：

依赖 jquery

**table 开发版本.js**：包含 ES6 代码，方便开发、调试，不兼容 IE8

**table.js**：开发版本经过 babel 转换得到的兼容版本，支持 IE8，但可读性较差

待改进：

loading 加载中效果

允许修改前后交接时，参数的名字

## 使用

### ➤ 初始化：

`var table1=new Table(param);` // param 是初始化参数，具体设置见下面“初始化参数”

### ➤ 刷新 refresh 方法

#### 1 有参数：有新的搜索条件

`var searchTerms = { name : “李” , age : 12};`

`table1.refresh ( search_terms );`

#### 2 无参数：没有新的搜索条件（沿用之前的搜索条件） 常用于 crud 之后重刷表格

`table1.refresh();`

### ➤ 获得选中数据（如果开启了单选、多选功能，具体见“初始化参数-选填”中的 selection 项）

`table1.get_selected( “sid” );`

## 前后交互说明

**url**: 初始化参数中设置的 url

前 -> 后：json 格式

```
{
  page:2,                // 当前是第几页
  number:20,             // 每页显示几条数据
  sort_name:'age',       // 排序列（如果需要排序的话）
  sort_method:'asc',     // 排序方式：'asc' 'desc'（如果需要排序的话）
  搜索条件 1: 搜索条件 1 的值,
  搜索条件 2: 搜索条件的值
}
```

后 -> 前：json 格式

```
{
  data: [ { json } , { json } ... ], // 数据
  count: 30                          // 数据总数
}
```

## 初始化参数 - 必填

- **id**      容器 id 类型 **string**
- **url**      ajax 请求数据的地址 类型：**string**
- **title**      标题行，具体见后面的“标题行参数” 类型：**[ {json}, {json} ... ]**

## 初始化参数 - 选填

- **searchTerms**  
搜索条件 类型：**{json}**，默认：**{}**
- **dataLimit**  
每页显示的条数，值务必对应 **dataLimits** 参数的选项 类型：**int**，默认：**10**
- **dataLimits**  
每页条数的选择项 类型：**[ int, int ... ]**，默认：**[5, 10, 20, 50]**
- **【排序功能】**：点击表头单元格，根据当前列将表格重新排序（只对开启了排序功能的列生效。如果想要对某一列 开启排序功能，将“标题行参数”中的 **sort** 项置为 **true**）  
**sort**: 'back' 类型：**string**，可选值：**back front**，默认 **front**  
    **back**：后台排序。需要后台支持，具体见后面的“前后交接参数说明”  
    **front**：前台 js 排序，只对当前页数据排序【默认值】
- **【序号功能】**：在最左侧添加一个序号列  
**auto\_index**:{  
    **show**: true,      // 是否启用此功能 类型：**boolean**，默认 **false**  
    **title**: '序号',      // 该列的标题 类型：**string**，默认 “序号”  
    **width**: 5      // 列宽(%) 类型：**int**，默认 **2**  
}
- **【单选多选功能】**：选中的数据行变色，可以通过 **get\_selected(key)**方法获取选中数据。  
    参数 **pid**：数据标识符  
        1 本组件使用 **pid[0]**来比较两条数据。比如下面的实例，通过对比两条数据的's-id'判断是否为同一条数据。  
        2 只有在 **pid** 中存取的字段，才可以通过 **get\_selected(key)**方法获取，比如下面的实例，可以使用 **get\_selected('s-id')**和 **get\_selected('x-sno')**  
    参数 **colomn\_shown**：是否显示按钮列  
        如果为 **true**，则会在表格左侧添加按钮列；为 **false**，则不显示按钮列（仍然有选中变色 效果，仍然可以使用 **get\_selected** 方法）  
    **selection**:{  
        **type**: 'checkbox',      // 选择方式 类型：**string**，可选值 **radio checkbox**，默认 “radio”  
        **pid**:['s-id','x-sno'],      // 数据标识符 类型：**[ string, string ... ]**

```
column_shown: true,    // 是否显示按钮列 类型: boolean, 默认 false
width: 5               // 列宽(%) 类型: int, 默认 2
}
```

➤ 【详情功能】: 在左侧添加一个 “+” 列, 点击+会显示当前行数据的更多信息

formatter 接受一个参数 index, 代表当前行数据是后台返回的 data 数据中的第几个。通过 table1.data[index] 可以取到当前行数据。

```
detail:{
  width:2,              // "+"列的列宽(%) 类型: int, 默认 2
  formatter:function(index){
    var data = table1.data[index]; // 取到当前行的数据
    return `

${data.name} : ${data.proverb}</p>`;
  }
}


```

## 标题行参数

- **name** (必填) 设定标题名称 类型 **string**
- **width** 设定列宽(%) 类型 **int**
- **key** (对于普通列是必填项) 设定字段名标题行 类型 **string**
- **sort** 是否允许对当前列排序 类型 **boolean**, 默认 **false**  
排序方式可以通过“初始化参数”中的 **sort** 项设置: 可以选择前台或后台排序
- **crud** 如果是操作列 (增删改查按钮), 必须把此项置为 **true** 类型 **boolean**, 默认 **false**
- **formatter** 类型 **function**

### 1 对于普通列:

需要设置 **key**, 并且设置 **crud** 为 **false** (或者不设置, **crud** 默认为 **false**)

**formatter** 的实参是 **key** 对应的值

### 2 对于操作列:

不需要设置 **key**, 并且设置 **crud** 为 **true**

**formatter** 的实参是 **index**, **index** 代表当前行对应数据是后台返回的 **data** 数据中的第几个。

通过 `table1.data[index]` 可以取到当前行数据。

例子: 对于数据 `{ sid: 12345, name: '小红', gender: 1 }`

普通列: **gender** 项 1 为男 其他为女

```
{
  name: '性别',
  key: 'gender',           // 必须设置 key
  formatter: function(gender) { // 参数是根据 key 来取到的
    return gender == "1" ? "男" : "女";
  }
}
```

操作列:

```
{
  name: '操作',
  crud: true,           // crud 必须置为 true
  formatter: function(index) { // 参数是 index
    var data = table1.data[index]; // 通过 table1.data[index] 可以取到当前行数据
    var sid = data.sid;
    return `<a title='详情' onclick='detail("${sid}")'>详情</a>`;
  }
}
```