# 配置选项

## data

data : Array of Arrays (default [ [] ]) //数据,支持 html 代码

data : Array of Objects

data : Function

## width & height

width : Number (default undefined) //宽度

width : Function //宽度

height : Number (default undefined) //高度

height : Function //高度

## Rows &Cols

minRows : Number (default 0) //初始化多少行

minCols : Number (default 0) //初始化多少列

maxRows : Number (default Infinity) //最多多少行

maxCols : Number (default Infinity) //最多多少列

startRows : Number (default 5) //开始多少行

startCols : Number (default 5) //开始多少列

## Headers

rowHeaders : Boolean (default false) //行标题 是否显示

rowHeaders : Array //行标题 可以是数组

colsHeaders : Boolean (default false) //列标题 是否显示

colsHeaders : Array //列标题 可以是数组

## 右键菜单

contextMenu : Boolean //右键菜单 支持数组事件已经写好

contextMenu: ['row\_above', 'row\_below', 'remove\_row']

## 列操作

manualColumnResize: true, //调整列宽

manualColumnMove: true, //移动列

## 冻结&滚动

scrollbarModelV: 'native' // 冻结头

stretchH: 'all',// 滚动条

fixedRowsTop: 2,

fixedColumnsLeft: 2, //冻结 行 和列

## 空行

minSpareRows: 1, //允许的空行

## 事件

fillHandle: true//是否允许拖拽

currentRowClassName: 'currentRow', //鼠标选中行效果

currentColClassName: 'currentCol', //鼠标选中列效果

columnSorting: true, //列排序

# 字段设置

## //字段种类

columns: [

{ data: "id", //字段名

readOnly: true , //只读

format: '$0,0.00', //数字格式化

type: 'handsontable', //字段 类型

validator: ip\_validator\_regexp , //表单验证

allowInvalid: true, //是否允许无效

handsontable: { colHeaders: false, data: colorData } //下拉菜单

},{},{},{},{},{}]

## //字段类型

  columns: [

{ type: 'text'}, //文本. [默认]

{ type: {renderer: yellowRenderer}}, //背景色

{ type: 'checkbox'}, //选择框

{ type: 'date'}, //日期

{ type: 'autocomplete'} //下拉菜单

{ type: 'numeric' } // 数字

# 操作

## //选择单元格

container.handsontable("selectCell", 0, 0);

## //设置属性

container.handsontable("updateSettings", {rowHeaders: $(**this**).is(':checked')});

container.handsontable("updateSettings", {colHeaders: $(**this**).is(':checked')});

## //样式处理

$('#example1 table').addClass('table'); //样式处理

$('.options input').on('change', **function** () {

**var** method = **this**.checked ? 'addClass' : 'removeClass';

$('#example1').find('table')[method](**this**.getAttribute('data-type'));

});

 table-bordered table-striped table-hover

## //事件

beforeKeyDown: **function** (e) { //

# 功能:

## //搜索字段

$('#search\_field').on('keyup',function(event) {

var value = ('' + this.value).toLowerCase(), row, col, r\_len, c\_len, td;

if (value) {

for (row = 0, r\_len = data.length; row < r\_len; row++) {

for (col = 0, c\_len = data[row].length; col < c\_len; col++) {

td = example.handsontable('getCell', row, col);

if (('' + data[row][col]).toLowerCase().indexOf(value) > -1) {

td.className = 'pass'

} else {

td.className = '';

}

}

}

}

});

## //下拉菜单

var colors = ["yellow", "red", "orange", "green", "blue", "gray", "black", "white"];

var colorData = [];

var color;

while (color = colors.shift()) {

colorData.push([[color]]);

}

columns: [ //字段种类

{

type: 'handsontable',

handsontable: {

colHeaders: false,

data: colorData

}

}]

## //表单验证

var ip\_validator\_regexp = /^(?:\b(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\b|null)$/;

columns: [{ validator: ip\_validator\_regexp},{},{},{},{}]

**var email\_validator\_fn = function(value, callback) {**

**if (/.+@.+/.test(value)) {**

**callback(true);**

**} else {**

**callback(false);**

**}**

**};**

**allowInvalid: true,**

**//callback 是否允许无效**

## Pagination //分页

# 事件

|  |  |  |
| --- | --- | --- |
| beforeInitWalkontable | Init Walkontable之前 | before Init Walkontable |
| beforeInit | 在Init | before Init |
| beforeRender | 渲染之前 | before Render |
| beforeChange | 更改前 | before Change |
| beforeValidate | 验证之前 | before Validate |
| beforeGet | 得到之前 | before Get |
| beforeSet | 设置之前 | before Set |
| beforeGetCellMeta | 得到细胞元之前 | before Get Cell Meta |
| beforeAutofill | 自动填补之前 | before Auto fill |
| beforeKeyDown | 键下之前 | before Key Down |
| afterInit | Init之后 | after Init |
| afterLoadData | 加载数据之后 | after Load Data |
| afterRender | 渲染之后 | after Render |
| afterChange | 更改之后 | after Change |
| afterValidate | 验证之后 | after Validate |
| afterGetCellMeta | 起床后细胞元 | after Get Cell Meta |
| afterGetColHeader | 得到列头之后 | after Get Col Header |
| afterGetColWidth | 得到列宽之后 | after Get Col Width |
| afterDestroy | 摧毁之后 | after Destroy |
| afterRemoveRow | 删除行之后 | after Remove Row |
| afterCreateRow | 创建行之后 | after Create Row |
| afterRemoveCol | 删除列之后 | after Remove Col |
| afterCreateCol | 创建列之后 | after Create Col |
| afterColumnResize | 在列调整之后 | after Column Resize |
| afterColumnMove | 在列移动之后 | after Column Move |
| afterDeselect | 取消选中之后 | after Deselect |
| afterSelection | 选择之后 | after Selection |
| afterSelectionByProp | 选择后通过道具 | after Selection By Prop |
| afterSelectionEnd | 选择结束后 | after Selection End |
| afterSelectionEndByProp | 选择结束后由支柱 | after Selection End By Prop |
| afterCopyLimit | 在复制限制 | after Copy Limit |
| modifyCol | 修改列 | modify Col |