

# Datatables的使用

datatables是一款基于JQuery的表格插件，主要用来优化table，支持表格分页、搜索、排序、显示条数、异步加载等众多好用的功能

项目地址：<https://datatables.net/>

## 基本使用

需要用到的JS和CSS文件位于项目代码下的media目录中，需要将这个目录中的对应文件放入你的项目里，这一步不赘述

1. 引入CSS/JS文件，由于Datatables是基于Jquery的，所以要先引入Jquery，这里我们都直接引入CDN的地址

```
<!-- 加载 JQuery -->
<script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>

<!-- 加载 Datatables -->
<link href="https://cdn.datatables.net/1.10.19/css/jquery.dataTables.min.css" rel="stylesheet" />
<script src="https://cdn.datatables.net/1.10.19/js/jquery.dataTables.min.js"></script>
```

1. 初始化Datatables

```
<table id="myTable" class="display" style="width:100%">
  <thead>
    <tr>
      <th>Name</th>
      <th>Position</th>
      <th>Office</th>
      <th>Age</th>
      <th>Start date</th>
      <th>Salary</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>0ps Coffee</td>
      <td>System Architect</td>
      <td>Edinburgh</td>
      <td>18</td>
      <td>2011/04/25</td>
      <td>$320,800</td>
    </tr>
    <!-- 中间省略一些tr数据 -->
    <tr>
      <td>Donna Snider</td>
      <td>Customer Support</td>
      <td>New York</td>
      <td>27</td>
      <td>2011/01/25</td>
      <td>$112,000</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <th>Name</th>
      <th>Position</th>
      <th>Office</th>
      <th>Age</th>
      <th>Start date</th>
      <th>Salary</th>
    </tr>
  </tfoot>
</table>
```

```
$(document).ready( function () {
    $('#myTable').DataTable();
});
```

1. 完成以上两步可以看到页面效果如下

Show 10 entries

Search:

Name	Position	Office	Age	Start date	Salary
Airi Satou	Accountant	Tokyo	33	2008/11/28	\$162,700
Angelica Ramos	Chief Executive Officer (CEO)	London	47	2009/10/09	\$1,200,000
Ashton Cox	Junior Technical Author	San Francisco	66	2009/01/12	\$86,000
Bradley Greer	Software Engineer	London	41	2012/10/13	\$132,000
Brenden Wagner	Software Engineer	San Francisco	28	2011/06/07	\$206,850
Brielle Williamson	Integration Specialist	New York	61	2012/12/02	\$372,000
Bruno Nash	Software Engineer	London	38	2011/05/03	\$163,500
Caesar Vance	Pre-Sales Support	New York	21	2011/12/12	\$106,450
Cara Stevens	Sales Assistant	New York	46	2011/12/06	\$145,600
Cedric Kelly	Senior Javascript Developer	Edinburgh	22	2012/03/29	\$433,060

Name

Position

Office

Age

Start date

Salary

Showing 1 to 10 of 57 entries

Previous

1

2

3

4

5

6

Next

## 功能开启/关闭

默认界面下除了原始的table外，还为table额外增加了搜索、列排序、分页及信息显示这些内容，如果不需要其中一个或多个可以通过以下设置隐藏

```
$( '#myTable' ).DataTable({
  "paging": false,
  "ordering": false,
  "info": false,
  "searching": false,
});
```

**paging**: 控制分页是否开启，默认开启，开启后会显示表格左上角的每页行数选择和右下角的页码跳转

**ordering**: 控制是否进行排序，默认开启，且默认会对第一列数据进行排序

**info**: 控制是否显示表格左下角的信息，默认开启

**searching**: 控制是否显示表格右上角的搜索，默认开启

## 排序配置

### 单列排序

可以通过order来设置初始排序的列以及排序方向

```
"order": [[ 3, "desc" ]],
```

列号默认从0开始算起，这里的3实际上对应的是第4列，这点要注意，以下所有用到列号的地方都是从0开始

注意：开启stateSave之后会导致columnDefs失效，两个参数尽量不要同时设置

### 多列排序

当然也可以在order里边配置同时按多列排序

```
"order": [[ 3, "desc" ], [ 0, "desc" ]],
```

如上配置的意思是先按照3列进行排序，如果3列相同，则再按照0列进行排序

## 搜索框提示设置

```
"fnPreDrawCallback": function( oSettings ) {
  $( '.dataTables_filter input' ).attr( { 'name': 'search', 'placeholder': '按名称搜索' } ); //提示
},
```

## 屏蔽错误提示和自定义错误提示

官方参考: <https://datatables.net/reference/event/error>

//初始化之前设置

```
$.fn.dataTable.ext.errMode = 'none'; //屏蔽报错弹窗
```

```
$( '#Table' ) //初始化的datatables
.on( 'error.dt', function ( e, settings, techNote, message ) {
    //接收并打印错误信息，也可自行添加相应处理方法
    console.log( 'An error has been reported by DataTables: ', message );
} ).DataTable({
    //其他配置
});
```

## 隐藏列

可以通过columnDefs来设置列属性

```
"columnDefs": [
    {
        "targets": [ 2 ],
        "visible": false,
        "searchable": false
    },
    {
        "targets": [ 3 ],
        "visible": false,
    }
]
```

**targets:** 指定列

**visible:** 是否可显示

**searchable:** 是否可搜索，当仅设置visible为false，但searchable不设置时这一列仍然可以被搜索

注意：开启stateSave之后会导致columnDefs失效，两个参数尽量不要同时设置

## 动态隐藏列

```
/**
 * #tagTable 表格id
 * column(5) 第6列
 * visible(false) true 显示 false不显示
 */
$( '#tagTable' ).DataTable().column(5).visible(false);
```

## 语言配置

默认提示语都是英文，可以通过language来设置为中文

```
"language": {
    "decimal": "",
    "emptyTable": "表中数据为空",
    "info": "显示第 _START_ 至 _END_ 项结果，共 _TOTAL_ 项",
    "infoEmpty": "显示第 0 至 0 项结果，共 0 项",
    "infoFiltered": "(由 _MAX_ 项结果过滤)",
    "InfoPostFix": "",
    "thousands": ",",
    "lengthMenu": "显示 _MENU_ 项结果",
    "loadingRecords": "载入中...",
    "processing": "处理中...",
    "search": "搜索:",
    "zeroRecords": "没有匹配结果",
    "Paginate": {
        "sFirst": "首页",
        "sPrevious": "上一页",
        "sNext": "下一页",
        "sLast": "尾页"
    },
    "Aria": {
        "sSortAscending": ": 以升序排列此列",
        "sSortDescending": ": 以降序排列此列"
    }
}
```

## 状态保持

可以通过stateSave来设置是否保持状态，stateSave会调用HTML5的localStorage和sessionStorage APIs将页面状态数据保存在浏览器本地，当你刷新页面时会自动加载这些状态信息，这些状态可以是你的排序信息、当前页码以及已输入的搜索数据等

```
"stateSave": true,
```

## 分页类型

可以通过pagingType来设置分页类型

```
"pagingType": "simple_numbers",
```

**simple\_numbers:** 类型会显示上一页、下一页按钮和页码，这也是默认的分页类型

其他几种分页类型如下：

**numbers:** 仅显示页码

**simple:** 仅显示上一页、下一页按钮

**full:** 仅显示首页、尾页、上一页和下一页按钮

**full\_numbers:** 显示首页、尾页、上一页、下一页按钮和页码

**first\_last\_numbers:** 显示首页、尾页按钮和页码

## 滚动配置

当table的宽、高超过页面设置大小时，可以通过以下配置来添加滚动条

```
"scrollX": "true",
"scrollY": "200px",
"scrollCollapse": true,
```

**scrollX:** 允许水平滚动条

**scrollY:** 设置垂直body的高度，当超过这个这个高度时出现垂直滚动条

**scrollCollapse:** 设置当数据占用高度小于scrollY设置的高度时自动收缩body高度

## 数据加载

上篇文章中的所有数据都是直接渲染的html中的table数据，datatables还支持其他几种数据源，以方便实现更灵活的控制

### 从数组中获取

```
<table id="myTable-x" class="display" style="width:100%"></table>
```

```
$(document).ready(function() {
    var dataSet = [
        ["3", "https://ops-coffee.cn", "2018-07-03"],
        ["9", "https://demo.ops-coffee.cn", "2019-08-06"],
    ];

    $('#myTable-x').DataTable({
        "data": dataSet,
        "columns": [
            { title: "Id" },
            { title: "Site" },
            { title: "Date" },
        ]
    })
});
```

**data:** 指定数组

**columns:** 配置每一列的title

注意：从数组中获取数据一定要有表头，如果没有则可能会报下边的错：

Uncaught TypeError: Cannot read property 'aDataSort' of undefined

解决方法就是datatables添加columns配置，或者写上table的thead

```
<table id="myTable-x" class="display" style="width:100%">
  <thead>
    <tr>
      <th>ID</th>
      <th>Site</th>
      <th>Date</th>
    </tr>
  </thead>
</table>
```

### 从对象中获取

```
<table id="myTable-x" class="display" style="width:100%"></table>
```

```
$(document).ready(function() {
    var dataSet = [
        {"Id": "3", "Site": "https://ops-coffee.cn", "Date": "2018-07-03"},
        {"Id": "9", "Site": "https://demo.ops-coffee.cn", "Date": "2019-08-06"},
    ];

    $('#myTable-x').DataTable({
        "data": dataSet,
        "columns": [
            {"data": "Id", "title": "Id"},
            {"data": "Site", "title": "Site"},
            {"data": "Date", "title": "Date"},
        ]
    })
});
```

使用对象数组，一定要配置columns的data，告诉DataTables每列对应的属性，title配置可选，添加title会给表格添加表头

### 从实例中获取

```
$(document).ready(function() {
    function dataSet(id, site, date) {
```

```

        this.id = id;
        this.site = site;
        this.date = date;
    };

    $('#myTable-x').dataTable({
        data: [
            new dataSet("3", "https://ops-coffee.cn", "2018-07-03"),
            new dataSet("9", "https://demo.ops-coffee.cn", "2019-08-06"),
        ],
        columns: [
            {"data": "id", "title": "Id"},
            {"data": "site", "title": "Site"},
            {"data": "date", "title": "Date"}
        ]
    });
});

```

## Ajax异步获取

Datatables还支持Ajax的方式异步加载数据，简单的方式是直接配置一个url地址即可

```

$(document).ready(function() {
    $('#myTable-x').dataTable({
        "ajax": 'sdata.json'
    });
});

```

ajax接收的数据可以是数组或者对象，注意columns的配置，可以对应参考前文两种格式数据的处理

Show  entries
 

Search:

Id	Site	Date
3	https://ops-coffee.cn	2018-07-03
9	https://demo.ops-coffee.cn	2019-08-06
9	https://ops-coffee.cn/t	2019-06-06
12	https://md2pdf.ops-coffee.cn	2019-07-05
15	https://demo.ops-coffee.cn/duallistbox	2019-08-06

Showing 1 to 5 of 7 entries
 

Previous
 
 2

## Ajax异步刷新

`$('#tagTable').DataTable().reload()` 直接刷新

`$('#tagTable').DataTable().ajax.url()` 支持新的url异步刷新

```

$('#tagTable').DataTable().ajax.url(' {url("index/material/listAll")}?searchClassId='+id).load();

```

## 表格重载

```

var table = $('#example').DataTable();
table.ajax.reload();

```

或者

```

$('#example').DataTable().ajax.reload();

```

## 当前页面重载

```

//重新加载当前页，且保持当前页码不变
$('#'+tableId).DataTable().draw(false);
true: 重置表格的初始状态，回到第一页，
false: 只是重新加载当前页，且保持当前页码不变。

```

## 列数据处理

上边的table可以发现有个site列的内容是一个网址，如果我们想让网址能够点击该如何实现呢？可以利用columns的render属性对展示结果进行更改

```

$(document).ready(function() {
    $('#myTable-x').dataTable({
        "ajax": 'sdata.json',
        "columns": [
            {"data": "id", "title": "Id"},
            {
                "data": "site",
                "title": "Site",
                "render": function (data, type, row) {
                    return '<a href='+data+' target="_blank">'+data+'</a>'
                }
            }
        ]
    });
});

```

```

    },
    { "data": "date", "title": "Date" }
  ]
});
});

```

render后边跟了个函数，每当数据表需要获取列中某个单元格的数据时render函数都会执行，且函数可能会被执行多次，函数默认接收三个参数，意思分别是：

**data**：单元格的具体数据，例如<https://ops-coffee.cn>

**type**：标识了这一次调用的请求类型，会有filter、display、type、sort

**row**：这一行的完整数据源，如果像Demo示例传了对象数据，那么可以通过row.site获取到这一行site列的数据

拿到参数进行一系列的处理后可以通过return返回最终想要展示的内容

当然也可以通过columns在表格末尾添加一列以实现编辑、删除的按钮展示

```

"columns": [
  { "data": "id", "title": "Id",
    {
      "data": "site",
      "title": "Site",
      "render": function (data, type, row) {
        return '<a href=' + data + ' target="_blank">' + data + '</a>'
      }
    }
  },
  { "data": "date", "title": "Date",
    {
      "data": "id",
      "title": "操作",
      "render": function (data, type, row) {
        return '<a href="#update/' + row.id + '/' + class="btn btn-warning btn-sm">编辑</a>' +
          '<a href="#delete/' + row.id + '/' + class="btn btn-danger btn-sm">删除</a>'
      }
    }
  }
]

```

最终呈现结果如下图

Show  entries
Search:

Id	Site	Date	操作
3	<a href="https://ops-coffee.cn">https://ops-coffee.cn</a>	2018-07-03	<button>编辑</button> <button>删除</button>
5	<a href="https://ops-coffee.cn/t">https://ops-coffee.cn/t</a>	2019-06-06	<button>编辑</button> <button>删除</button>
9	<a href="https://demo.ops-coffee.cn">https://demo.ops-coffee.cn</a>	2019-08-06	<button>编辑</button> <button>删除</button>
12	<a href="https://md2pdf.ops-coffee.cn">https://md2pdf.ops-coffee.cn</a>	2019-07-05	<button>编辑</button> <button>删除</button>
15	<a href="https://demo.ops-coffee.cn/duallistbox">https://demo.ops-coffee.cn/duallistbox</a>	2019-08-06	<button>编辑</button> <button>删除</button>

Showing 1 to 5 of 7 entries

Previous
1
2
↑

## Dom操作

如果我不需要datatables显示左上角的每页显示条数信息，而要换成一个添加按钮改怎么做呢？这里可以借助datatables的dom来实现

默认情况下表格都会有左上角的每页显示条数、右上角的搜索、左下角的表格信息、右下角的分页、中间的数据加载等待以及表格本身，这些都是datatables的DOM，它们实际上就是一个div包裹起来的select、input之类的html标签，datatables中的每个DOM都与一个字母相对应，他们的对应关系如下：

**l**：length，代表左上角的每页显示条数控件

**f**：filtering，代表右上角的搜索控件

**t**：table，代表表格本身

**i**：information，代表左下角的表格信息控件

**p**：pagination，代表右下角的分页控件

**r**：processing，代表中间数据加载等待提示控件

这些控件在datatables里可以通过配置dom来控制他们的显示位置，以及是否显示，默认的显示顺序是lfrtip

```

$('#myTable-x').dataTable({
  "dom": 'lfrtip'
})

```

你如果不想显示某个控件，可以通过去掉`dom`配置项里对应的字母实现，同时Datatables支持四个自定义的标签，通过这四个标签可以方便的来修改DOM的展示

< > 尖括号就代表html里的div

<"class"> 代表了添加了class的div

<"#id"> 代表了添加了id的div

<"#id.class"> 代表添加了id和class的div

我们想把右上角的每页显示条数控件换成添加按钮的话可以这样写

```
$('#myTable-x').dataTable({
  "dom": '<"#add-btn.toolbar">frtip'
})

$("#add-btn.toolbar").html(
  '<button href="#add" class="btn btn-success btn-sm"> + 添加</button>'
)
```

遇到样式问题，需要添加css

```
<style type="text/css">
  .toolbar {float:left}
</style>
```

这样就完美实现了

+ 添加

Search:

Id	Site	Date	操作
3	<a href="https://ops-coffee.cn">https://ops-coffee.cn</a>	2018-07-03	<div>编辑删除</div>
5	<a href="https://ops-coffee.cn/t">https://ops-coffee.cn/t</a>	2019-06-06	<div>编辑删除</div>
9	<a href="https://demo.ops-coffee.cn">https://demo.ops-coffee.cn</a>	2019-08-06	<div>编辑删除</div>
12	<a href="https://md2pdf.ops-coffee.cn">https://md2pdf.ops-coffee.cn</a>	2019-07-05	<div>编辑删除</div>
15	<a href="https://demo.ops-coffee.cn/duallistbox">https://demo.ops-coffee.cn/duallistbox</a>	2019-08-06	<div>编辑删除</div>

Showing 1 to 5 of 7 entries

Previous

1

2

## 服务器端处理

Datatables支持使用服务端进行数据处理，当开启服务端数据处理后，Datatables将在页面执行分页、排序、搜索等操作时向服务端发出Ajax请求，Ajax请求会传递许多变量给服务端，服务端接收到请求后根据变量的值对数据进行处理，处理完成按照固定的格式返回给前端页面，页面对返回的数据进行渲染提供给用户查看

开启服务器模式只需要两个设置项`serverSide`和`ajax`

```
$('#myTable-x').dataTable({
  "serverSide": true,
  "processing": true,
  "ajax": '/api/site/data'
})
```

**serverSide:** 为true时表示开启服务端处理模式

**processing:** 为true时会开启数据处理中的提示，非必须

**ajax:** 指定服务器端的地址，可以像上边一样是个字符串，也可以像jQuery.ajax一样作为一个对象使用，例如我想传递额外的参数（datatables默认会给后端传递许多的参数，下边有讲）给后端服务器的话，可以这样用

```
$('#myTable-x').dataTable({
  "serverSide": true,
  "processing": true,
  "ajax": {
    "url": "/api/site/data",
    "data": function (d) {
      d.type = 'ops-coffee';
    }
  }
})
```

**data:** 可以在发送请求给后端时额外增加`type=ops-coffee`的参数

## 发送到服务器端的参数

当开启服务端数据处理后，默认会给服务端传递许多参数，大概如下：

**draw**: 绘制计数器，主要用来确保Ajax从服务器端接收到的数据是对应同一次请求的  
**start**: 第一条数据的起始位置  
**length**: 每页显示的条数  
**search[value]**: 全局的检索关键字  
**order[i][column]**: 告诉服务器哪些列是需要排序的，i为排序列的序号，下边的i相同含义，注意i是从0开始的  
**order[i][dir]**: 告诉服务器排序的方式“desc”, “asc”  
**columns[i][data]**: columns上定义的data属性值  
**columns[i][name]**: columns上定义的name属性值  
**columns[i][searchable]**: 告诉服务器哪些列可以被搜索  
**columns[i][orderable]**: 告诉服务器哪些列可以进行排序  
**columns[i][search][value]**: 告诉服务器某些列的具体搜索条件

如果需要后台分页，那么需要拿到start、length两个参数做相应的处理，

如果有搜索的内容，那么需要拿到serch[value]参数做处理

## 服务端返回数据的格式

服务端需要返回datatables可以处理的数据格式，具体数据格式如下：

```
{
  "draw": 1,
  "recordsTotal": 7,
  "recordsFiltered": 7,
  "data": [
    {
      "id": 3,
      "site": "https://ops-coffee.cn",
      "date": "2018-07-03"
    },
    {
      "id": 9,
      "site": "https://demo.ops-coffee.cn",
      "date": "2019-08-06"
    }
  ]
}
```

**draw**: 客户端调用服务端次数标识，客户端传过来是什么原样返回回去即可，无需修改

**recordsTotal**: 数据总条数，没有过滤的数据总条数

**recordsFiltered**: 过滤后符合要求的条数，如果没有搜索参数那么这个值与recordsTotal一致

**data**: 需要显示的具体数据，json格式

## API调用

Datatables提供了强大的API来处理表格上的数据，可以通过API添加数据到已经存在的表格，或者对已经存在的数据进行操作，API的类型非常丰富，详细的信息可以查阅官网，使用方法如下：

### 跳转到页

跳转到第3页：

```
var table = $('#myTable').DataTable()

table.page(2).draw(false)
```

**page(2)**: page为分页方法，后边的2表示跳转到第几页，可以是一个数字，也可以是first、next、previous、last这样的字符串，当为数字时要从0算起，例如示例中为2实际上是跳转到了第3页

**draw(false)**: 对表格进行重绘以实现表格更新的显示，大多数的api操作都不会直接更新在页面上，所以需要调用下draw，默认情况下重绘后分页会被重置回到第一页，当设置为false时分页不会被重置

### 搜索某列

搜索第2列包含https://ops-coffee.cn的行

```
var tablx = $('#myTable').dataTable()

tablx.api().column(1).search('https://ops-coffee.cn').draw()
```

首先需要注意这个例子中的API调用使用了.api()，这是因为上一个例子在初始化时用了.DataTable()而这个例子初始化时用了.dataTable()，仅仅是d字母大小写的区别而已，但意义确不同，前者直接返回API实例，后者返回的是jQuery实例

### 重新设置ajax data属性

```
var api = $('#tagTable').dataTable().api();
//初始化的ajax data选项
$data = api.settings()[0].ajax.data;
//重新设置
$data.type = '12321321';
//重载表格
api.draw();
```



# 导出

<https://datatables.net/download> 官方下载

## 导出Excel

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>tableTools</title>
  <!-- jQuery -->
  <script type="text/javascript" charset="utf8" src="http://code.jquery.com/jquery-1.10.2.min.js"></script>

  <!-- datatables 的样式和导出excel的js-->
  <link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/v/dt/jszip-2.5.0/dt-1.10.21/b-1.6.2/b-flash-1.6.2/b-html5-1.6.2/datatables.min.js">
  <script type="text/javascript" src="https://cdn.datatables.net/v/dt/jszip-2.5.0/dt-1.10.21/b-1.6.2/b-flash-1.6.2/b-html5-1.6.2/datatables.min.js">
  <script>
    $(document).ready(function () {
      $('#myTable').DataTable({
        dom: 'Bfrtip',
        buttons: [
          'excelHtml5',
        ]
      });
    });
  </script>
</head>
<body>
<div class="col-md-12 col-sm-12">
  <!-- BEGIN EXAMPLE TABLE PORTLET-->
  <div class="portlet box blue-hoki">
    <div class="portlet-title">
      <div class="caption">
        <i class="fa fa-list"></i>数据列表
      </div>
      <div class="tools">
      </div>
    </div>
    <div class="portlet-body">
      <table id="myTable" class="table table-striped table-bordered table-hover"
        style="width:100%">
        <thead>
          <tr>
            <th>Name</th>
            <th>Position</th>
            <th>Office</th>
            <th>Age</th>
            <th>Start date</th>
            <th>Salary</th>
          </tr>
          <tr>
            <th>Name111</th>
            <th>Position111</th>
            <th>Office111</th>
            <th>Age111</th>
            <th>Start date111</th>
            <th>Salary111</th>
          </tr>
          </thead>
          <tbody>

            <tr>
              <td>Airi Satou</td>
              <td>Accountant</td>
              <td>Tokyo</td>
              <td>33</td>
              <td>2008/11/28</td>
              <td>$162,700</td>
            </tr>
            <tr>
              <td>Brielle Williamson</td>
              <td>Integration Specialist</td>
              <td>New York</td>
              <td>61</td>
              <td>2012/12/02</td>
              <td>$372,000</td>
            </tr>
            <tr>
              <td>Herrod Chandler</td>
              <td>Sales Assistant</td>
              <td>San Francisco</td>
              <td>59</td>
              <td>2012/08/06</td>
              <td>$137,500</td>
            </tr>
            <tr>
              <td>Jonas Alexander</td>
              <td>Developer</td>
              <td>San Francisco</td>
              <td>30</td>
              <td>2010/07/14</td>
              <td>$86,500</td>
            </tr>
            <tr>
              <td>Shad Decker</td>
              <td>Regional Director</td>
              <td>Edinburgh</td>
              <td>51</td>
              <td>2008/11/13</td>
            </tr>
          </tbody>
        </table>
      </div>
    </div>
  </div>
</div>
```

```
<td>$183,000</td>
</tr>
<tr>
<td>Michael Bruce</td>
<td>Javascript Developer</td>
<td>Singapore</td>
<td>29</td>
<td>2011/06/27</td>
<td>$183,000</td>
</tr>
<tr>
<td>Donna Snider</td>
<td>Customer Support</td>
<td>New York</td>
<td>27</td>
<td>2011/01/25</td>
<td>$112,000</td>
</tr>
</tbody>
<tfoot>
<tr>
<th>Name</th>
<th>Position</th>
<th>Office</th>
<th>Age</th>
<th>Start date</th>
<th>Salary</th>
</tr>
</tfoot>
</table>
</div>
</div>
</body>
</html>
```

运行结果

数据列表

Excel

Name	Position	Office	Age
Name111	Position111	Office111	Age11
Airi Satou	Accountant	Tokyo	33
Brielle Williamson	Integration Specialist	New York	61
Donna Snider	Customer Support	New York	27
Herrod Chandler	Sales Assistant	San Francisco	59
Jonas Alexander	Developer	San Francisco	30
Michael Bruce	Javascript Developer	Singapore	29
Shad Decker	Regional Director	Edinburgh	51

NamePositionOfficeAge

Showing 1 to 7 of 7 entries

tableTools (2).xlsx

## 导出PDF

pdf是不支持中文的暂时无解决方案

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```

<meta charset="UTF-8">
<title>tableTools</title>
<!-- jQuery -->
<script type="text/javascript" charset="utf8" src="http://code.jquery.com/jquery-1.10.2.min.js"></script>

<!--datatables 的样式和导出excel的js-->
<link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/v/dt/jszip-2.5.0/dt-1.10.21/b-1.6.2/b-flash-1.6.2/b-html5-1.6.2/datatables.min.js">
<script type="text/javascript" src="https://cdn.datatables.net/v/dt/jszip-2.5.0/dt-1.10.21/b-1.6.2/b-flash-1.6.2/b-html5-1.6.2/datatables.min.js">

<!--PDF-->
<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.36/pdfmake.min.js"></script>
<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.36/vfs_fonts.js"></script>
<script type="text/javascript" src="https://cdn.datatables.net/v/dt/dt-1.10.21/b-1.6.2/b-html5-1.6.2/datatables.min.js"></script>

<script>
    $(document).ready(function () {
        $('#myTable').DataTable({
            dom: 'Bfrtip',
            buttons: [
                'excelHtml5',
                'pdfHtml5',
            ]
        });
    });
</script>
</head>
<body>
<div class="col-md-12 col-sm-12">
<!-- BEGIN EXAMPLE TABLE PORTLET-->
<div class="portlet box blue-hoki">
    <div class="portlet-title">
        <div class="caption">
            <i class="fa fa-list"></i>数据列表
        </div>
        <div class="tools">
        </div>
    </div>
    <div class="portlet-body">
        <table id="myTable" class="table table-striped table-bordered table-hover"
            style="width:100%">
            <thead>
                <tr>
                    <th>Name</th>
                    <th>Position</th>
                    <th>Office</th>
                    <th>Age</th>
                    <th>Start date</th>
                    <th>Salary</th>
                </tr>
                <tr>
                    <th>Name111</th>
                    <th>Position111</th>
                    <th>Office111</th>
                    <th>Age111</th>
                    <th>Start date111</th>
                    <th>Salary111</th>
                </tr>
            </thead>
            <tbody>

                <tr>
                    <td>Airi Satou</td>
                    <td>Accountant</td>
                    <td>Tokyo</td>
                    <td>33</td>
                    <td>2008/11/28</td>
                    <td>$162,700</td>
                </tr>
                <tr>
                    <td>Brielle Williamson</td>
                    <td>Integration Specialist</td>
                    <td>New York</td>
                    <td>61</td>
                    <td>2012/12/02</td>
                    <td>$372,000</td>
                </tr>
                <tr>
                    <td>Herrod Chandler</td>
                    <td>Sales Assistant</td>
                    <td>San Francisco</td>
                    <td>59</td>
                    <td>2012/08/06</td>
                    <td>$137,500</td>
                </tr>
                <tr>
                    <td>Jonas Alexander</td>
                    <td>Developer</td>
                    <td>San Francisco</td>
                    <td>30</td>
                    <td>2010/07/14</td>
                    <td>$86,500</td>
                </tr>
                <tr>
                    <td>Shad Decker</td>
                    <td>Regional Director</td>
                    <td>Edinburgh</td>
                    <td>51</td>
                    <td>2008/11/13</td>
                    <td>$183,000</td>
                </tr>
            </tbody>
        </table>
    </div>
</div>

```

```
<td>Michael Bruce</td>
<td>Javascript Developer</td>
<td>Singapore</td>
<td>29</td>
<td>2011/06/27</td>
<td>$183,000</td>
</tr>
<tr>
<td>Donna Snider</td>
<td>Customer Support</td>
<td>New York</td>
<td>27</td>
<td>2011/01/25</td>
<td>$112,000</td>
</tr>
</tbody>
<tfoot>
<tr>
<th>Name</th>
<th>Position</th>
<th>Office</th>
<th>Age</th>
<th>Start date</th>
<th>Salary</th>
</tr>
</tfoot>
</table>
</div>
</div>
</body>
</html>
```

运行效果

数据列表

ExcelPDF

Name	Position	Office
Name111	Position111	Office111
Airi Satou	Accountant	Tokyo
Brielle Williamson	Integration Specialist	New York
Donna Snider	Customer Support	New York
Herrod Chandler	Sales Assistant	San Francisco
Jonas Alexander	Developer	San Francisco
Michael Bruce	Javascript Developer	Singapore
Shad Decker	Regional Director	Edinburgh

NamePositionOffice

Showing 1 to 7 of 7 entries

tableTools (1).pdf  
14.2/14.2 KB

## 导出按钮的配置

例:

```
dom: 'Bfrrtip',
buttons: [
  {
    //按钮
    extend: "copy",
    //按钮名称
    text: '复制',
    //按钮的class名称
    className : 'export-btn'
  },
  {
    extend: "excelHtml5",
    text: '导出excel',
    className : 'export-btn'
  },
  {
    extend: "copy",
    text: '导出pdf',
    className : 'export-btn'
  },
  // 'copy', 'excelHtml5', 'pdfHtml5'
],
```

### 所有配置选项

<https://datatables.net/reference/option/buttons.buttons>

- `buttons.buttons.action` -描述激活动作的功能
- `buttons.buttons.className` -按钮类名称
- `buttons.buttons.enabled` -初始启用状态
- `buttons.buttons.extend` -基于扩展对象
- `buttons.buttons.init` -按钮初始化回调函数
- `buttons.buttons.key` -按键激活配置
- `buttons.buttons.name` -在选择器中使用的按钮名称
- `buttons.buttons.text` -可见文字

## 导出设置

<https://datatables.net/extensions/buttons/examples/html5/outputFormat-function.html>

参考文章: <https://ops-coffee.cn/s/mx2gjLmO7jHKagGsWF6eJw>

## 事件相关

<http://datatables.club/reference/event/>

Datatables对自己所触发的事件都有监听，我们可以监听这些动作做相应的处理，比如表格重绘的时候，我需要做一些操作

Datatables的事件监听使用 `on()` API 方法或者 `jQuery.on()` ,事件的命名都用dt结尾， 保证和其他的命名不冲突，下面的例子显示了如何监听 `drawEvent` 事件

```
$( '#myTable' ).on( 'draw.dt', function() {
    alert( 'Table redrawn' );
});
```

## 事件(Events)

名称	说明
<code>errorEvent</code>	错误事件-当在加载数据发生错误时
<code>column-sizingDT</code>	列大小事件-当表格的列宽重新计算
<code>column-visibilityDT</code>	列显示事件-当表格列显示或者隐藏
<code>destroyDT</code>	表格销毁事件-当表格被销毁时
<code>drawDT</code>	重绘事件-当表格重绘完成后
<code>initDT</code>	加载完成事件-当表格完全加载完毕后

名称	说明
lengthDT	分页长度改变事件-当分页长度发生改变
orderDT	排序事件-当表格里数据发生排序
pageDT	页面改变事件-当页面发生改变
preXHRDT	ajax事件-当datatable发出ajax请求前
processingDT	处理过程事件-当表格正在处理（排序，或者加载）时
searchDT	搜索事件-当过滤数据时
stateLoadedDT	状态被加载事件-一旦状态已经被加载应用
stateLoadParamsDT	状态加载事件-当状态正在加载时
stateSaveParamsDT	状态存储事件-当正在保存表格状态信息时
xhrDT	ajax事件-当datatable发送ajax请求完成时