jQuery-DataTabels

1.HelloWorld示例

<!--第一步：引入Javascript / CSS （CDN）-->

<!-- DataTables CSS -->

<link rel="stylesheet" type="text/css" href="http://cdn.datatables.net/1.10.13/css/jquery.dataTables.css">

<!-- jQuery -->

<script type="text/javascript" charset="utf8" src="http://code.jquery.com/jquery-1.10.2.min.js"></script>

<!-- DataTables -->

<script type="text/javascript" charset="utf8" src="http://cdn.datatables.net/1.10.13/js/jquery.dataTables.js"></script>

<!--第二步：添加如下 HTML 代码-->

<table id="table\_id\_example" class="display">

<thead>

<tr>

<th>Column 1</th>

<th>Column 2</th>

</tr>

</thead>

<tbody>

<tr>

<td>Row 1 Data 1</td>

<td>Row 1 Data 2</td>

</tr>

<tr>

<td>Row 2 Data 1</td>

<td>Row 2 Data 2</td>

</tr>

</tbody>

</table>

<script type="text/javascript">

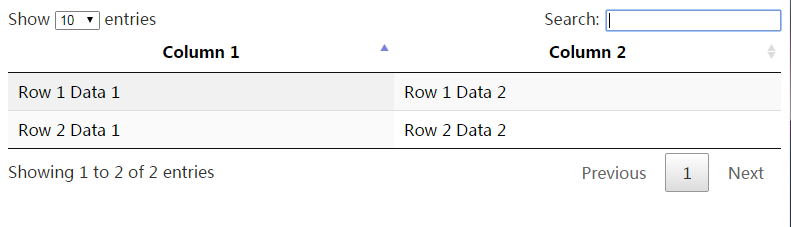
<!--第三步：初始化Datatables-->

$(document).ready( function () {

$('#table\_id\_example').DataTable();

} );

</script>



可以看出，自带分页展示和前端内容搜索功能

2. 简单的使用 DataTables 可能无法达到你的要求，这就需要你查找更多的选项来加强你的table：

* 定制你需要的 DataTables [选项](http://datatables.club/reference/option)
* 使用 DataTables 提供的[API](http://datatables.club/reference/api)来控制你的表格
* 定制适合自己网站的[样式](http://datatables.net/manual/styling/theme-creator)给 DataTables
* 查看 DataTables 的[示例](http://datatables.club/example)（**这个是最重要的**）

3.定制选项 Option

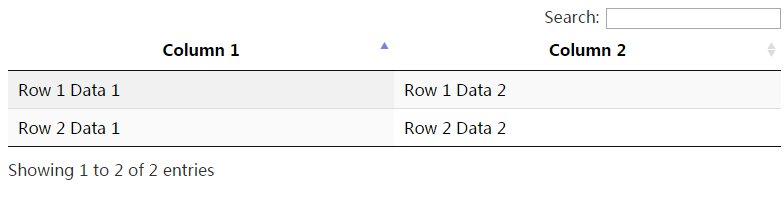
1） [scrollYOption](http://datatables.club/reference/option/scrollY.html)和 [pagingOption](http://datatables.club/reference/option/paging.html)选项用来**允许滚动**和禁止分页：

$('#myTable').DataTable( {

scrollY: 300, //允许y方向滚动

paging: false

} );



2）datatables的配置是通过设置你定义的选项来完成的，如下：

|  |  |
| --- | --- |
| 1  2  3 | $('#example').DataTable( {      paging: false  } ); |

通过设置paging选项，禁止表格分页(默认是打开的)

假设你要在表格里使用滚动，你需要加上scrollY选项：

|  |  |
| --- | --- |
| 1  2  3 | $('#example').DataTable( {      scrollY: 400  } ); |

当然你可以组合多个选项来初始化datatables，启动滚动条，禁用分页

|  |  |
| --- | --- |
| 1  2  3  4 | $('#example').DataTable( {      paging: false,      scrollY: 400  } ); |

### 3）常用选项(Common options)

下面列举了一些比较有用的选项:

* ajax - 异步数据源配置
* data - javascript数据源配置
* serverSide - 开启服务器模式
* columns.data - 配置每一列的数据源
* scrollX - 水平滚动条
* scrollY - 垂直滚动条

# 4. Ajax

Data for a DataTable can essentially come from three different locations:

为DataTable准备的数据可以来自三个位置

* [The HTML](https://datatables.net/manual/data#DOM) – 一个来自HTML中已存在并且已有数据的table
* [A Javascript array](https://datatables.net/manual/data#Javascript) – 一个数据数组
* An Ajax data source— 一个Ajax数据源

我们就来谈谈Ajax数据源

1）DataTable加载数据源的基本形式

通过使用ajax设置url，该url指向Ajax源数据

$('#myTable').DataTable( {

     ajax: '/api/myData'

} );

2）It is possible to use other data formats such as [XML](https://en.wikipedia.org/wiki/XML) and [YAML](https://en.wikipedia.org/wiki/YAML) with DataTables, although these formats need to be converted to Javascript object notation (i.e. JSON) before they are using - this is typically done in using [ajax.dataSrc](https://datatables.net/reference/option/ajax.dataSrc). The remainder of this document will consider only JSON.

我们不一定要使用JSON数据源，使用XML等也是可以的，但是要使用ajax.datasrc来引用，因为DataTable设定最终要转化成JSON数据

3）源数据位置

DataTable需要的源数据要以数组的形式存在

the first thing we need to do is tell DataTables where that array is in the data source

然后，The [ajax.dataSrc](https://datatables.net/reference/option/ajax.dataSrc) (i.e. data source) option is used to tell DataTables **where the data array is** in the JSON structure.

(1)简单的json数组ajax加载示例

[

    "Tiger Nixon",

    "System Architect",

    "Edinburgh",

    "5421",

    "2011/04/25",

    "$3,120"

]

$('#myTable').DataTable( {

    ajax: {

        url: '/api/myData',

        dataSrc: ''

    },

    columns: [ ... ]

} );

（2）Object with data property - note that the data parameter format shown here can be used with a simplified DataTables initialisation as data is the default property that DataTables looks for in the source data object.

Data属性是DataTable用来寻找数据源的默认属性

{

    "data": [

        {

            "name": "Tiger Nixon",

            "position": "System Architect",

            "salary": "$320,800",

            "start\_date": "2011/04/25",

            "office": "Edinburgh",

            "extn": "5421"

        },

        ...

    ]

}

$('#myTable').DataTable( {

    ajax: '/api/myData', //JSON数据源

    columns: [ ... ] //由于data属性是DataTable的默认寻找属性，所以这里不用**datasrc属性来指明json数据源中的数组**

} );

// or!

$('#myTable').DataTable( {

    ajax: {

        url: '/api/myData',

        dataSrc: 'data'

    },

    columns: [ ... ]

} );

我们可以对比使用其他属性时，比如随便一个：staff

{

    "staff": [

        {

            "name": "Tiger Nixon",

            "position": "System Architect",

            "salary": "$320,800",

            "start\_date": "2011/04/25",

            "office": "Edinburgh",

            "extn": "5421"

        },

        ...

    ]

}

$('#myTable').DataTable( {

    ajax: {

        url: '/api/myData',

        dataSrc: 'staff'

    },

    columns: [ ... ]

} );

此时 我们必须要使用datasrc属性来指明json数据源中的对应数组是哪个

## 4）Column data points

Now that DataTables knows where to get the data for the rows, we need to also tell it where to get the data for each cell in that row - this is done through the [columns.data](https://datatables.net/reference/option/columns.data) option.

既然DataTable知道从哪里为行获取数据，我们还要告诉他把数据填入行中的哪个位置，这通过DataTable的columns属性来完成

1. 数据数组（一个数组代表一行数据）

{

[

    "Tiger Nixon",

    "System Architect",

    "$320,800",

    "2011/04/25",

    "Edinburgh",

    "5421"

]

…..

[

……

]

}

我们可以直接加载源数据

$('#myTable').DataTable( {

    ajax: ...

} );

// or! 或者指明行对应数组中的index

$('#myTable').DataTable( {

    ajax: ...,

    columns: [

        { data: 0 },

        { data: 1 },

        { data: 2 },

        { data: 3 },

        { data: 4 },

        { data: 5 }

    ]

} );

1. 数据对象Object of data:

在这里我们先复习一下JSON格式的定义：

1） 并列的数据之间用逗号（”, “）分隔。  
  
2） 映射用冒号（”: “）表示。  
  
3） 并列数据的集合（数组）用方括号(“[]“)表示。  
  
4） 映射的集合（对象）用大括号（”{}”）表示。

这是一个对象，这里作为数据源传入，需要columns属性对应对象的属性名

{

    "name": "Tiger Nixon",

    "position": "System Architect",

    "salary": "$320,800",

    "start\_date": "2011/04/25",

    "office": "Edinburgh",

    "extn": "5421"

}

$('#myTable').DataTable( {

    ajax: ...,

    columns: [

        { data: 'name' },

        { data: 'position' },

        { data: 'salary' },

        { data: 'state\_date' },

        { data: 'office' },

        { data: 'extn' }

    ]

} );

1. 嵌套对象Nested objects

in this case note that in the nested objects we use dotted object notation such as hr.position to access nested data. With this ability almost any JSON data structure can be used with DataTables:

在这种情况下，我们使用 . 对象来引用嵌套的数据，这样几乎所有的JSON数据结构都能被用在DataTables上

{

    "name": "Tiger Nixon",

    "hr": {

        "position": "System Architect",

        "salary": "$320,800",

        "start\_date": "2011/04/25"

    },

    "contact": {

        "office": "Edinburgh",

        "extn": "5421"

    }

}

$('#myTable').DataTable( {

    ajax: ...,

    columns: [

        { data: 'name' },

        { data: 'hr.position' }, //子属性引用

        { data: 'hr.salary' },

        { data: 'hr.state\_date' },

        { data: 'contact.office' },

        { data: 'contact.extn' }

    ]

} );

更多例子展示：https://datatables.net/examples/ajax/

更多Option：<http://datatables.club/reference/option/>

自助练习：<http://datatables.club/example/diy.html>

5.