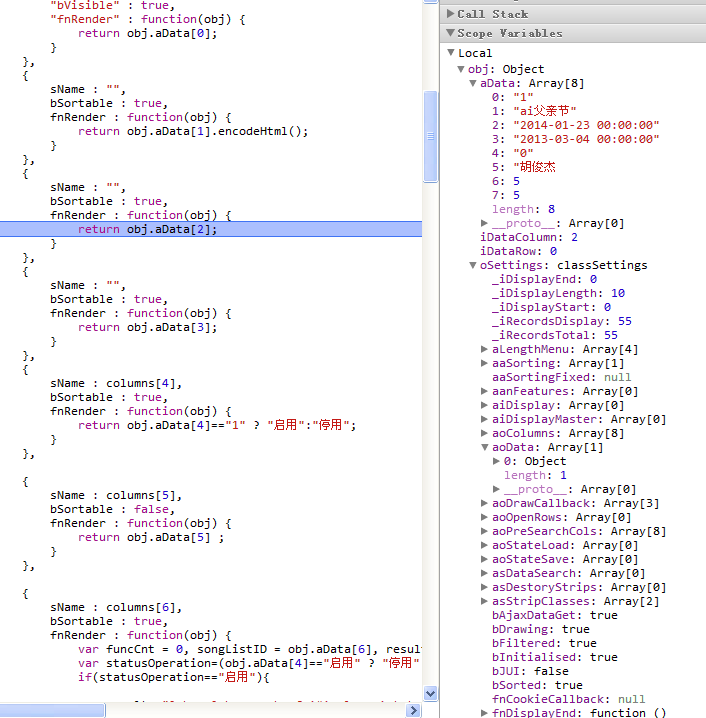
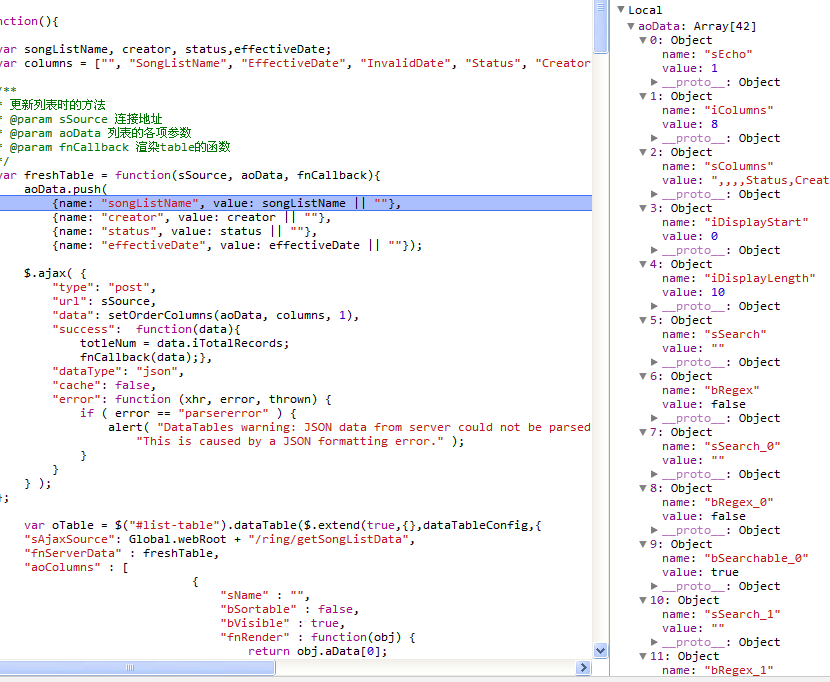
**data Tables的使用方法**

"bVisible" : **false**,不显示列

**1，bLengthChange 是否显示选择table中显示数据量 show(10,20,100...)column  
2，bPaginate 是否显示分页menus  
3，bProcessing载入数据时，是否显示‘进度’提示  
4，bInfo  是否显示有关table中现有数据的信息，执行过滤数据操作(filtered data)时显示对应信息  
5，bAutoWidth  
是否启动自动计算各个字段(td单元格)所在宽度  
   
$(document).ready( function () {  
  $('#example').dataTable( {  
    "bAutoWidth": false  
  } );  
} );  
6，bScrollInfinite 是否启动初始化滚动条--数据过多了，通常和sScrollY结合使用  
不能和pagination分页功能并存，分页了数据就不会过多了(垂直方向）  
$(document).ready( function() {  
  $('#example').dataTable( {  
    "bScrollInfinite": true,  
    "bScrollCollapse": true,  
    "sScrollY": "200px"  
  } );  
} );  
   
7，bServerSide是否启动服务器端数据导入，即要和****sAjaxSource结合使用  
  $(document).ready( function () {  
    $('#example').dataTable( {  
      "bServerSide": true,  
      "sAjaxSource": "xhr.**[**php**](http://www.2cto.com/kf/web/php/)**"  
    } );  
  } );  
   
8，bSort 是否启动各个字段的排序功能--点击table对应字段的最上面一下，  
    该列就会进行asc或desc排序  
   bSortable可对应启动单个字段的排序功能  
   
9，bSortClasses 是否启动给正在被排序的列(字段)加上class(有sorting\_1,sorting\_2和      
   sorting\_3三种)属性，以凸显正在排序的列，这个功能耗时间，数据大时，关闭  
   
10，bStateSave 是否启动状态保留，开启cookie记住table信息如分页信息，  
   数据长度，filtering过滤和sorting过滤等信息。  
   
11，sScrollX 字段过多，启动横向滚动条  
  $(document).ready( function() {  
  $('#example').dataTable( {  
    "sScrollX": "100%",  
    "bScrollCollapse": true  
  } );  
} );  
12,sScrollY  数据过多，使用垂直滚动条 第六个有使用到  
   $(document).ready( function() {  
  $('#example').dataTable( {  
    "sScrollY": "200px",  
    "bPaginate": false  
  } );  
} );  
   
13，bFilter 是否启动搜索功能  
   
14，bDestroy  
    $(document).ready( function() {  
      $('#example').dataTable( {  
        "sScrollY": "200px",     //显示200px垂直长度窗口  
        "bPaginate": false       //禁用分页  
       } );  
    
     // Some time later....  
      $('#example').dataTable( {  
         "bFilter": false,  
         "bDestroy": true        //这使得在200px长度窗口显示出来的数据，下一个窗口显示  
                                 //有使得有了分页功能  
       } );  
   } );  
15,** **iDisplayLength 一个分页显示多少行数据  
16, iDisplayStart 从第几行开始显示数据  
   
18，sPaginationType  
  DataTables内建方法有‘two\_button’和‘full\_numbers’  
     
$(document).ready( function() {  
  $('#example').dataTable( {  
    "sPaginationType": "full\_numbers"  
  } );  
} )  
这类方法是可以自定义的，如：自己写个listbox方法，提供一个下拉框来选择显示当前一个table中显示多少数据，  
$(document).ready(function() {  
  $('#example').dataTable( {  
     "sPaginationType": "listbox"  
   } );  
} );**

var oTable = $("#list-table").dataTable($.extend(true,{},dataTableConfig,{})；

oTable是一个自定义配置的datatables对象，其内部包含许多插件自带的函数，如fnAddData()、fnDraw(),fnSort(),fnFilter()…可直接通过oTable.fnAddData()调用相关函数，obj为后台返回的对象，其属性如下：

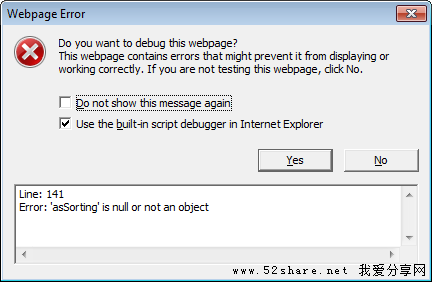
**  
 **

**aoData中包含datatables的一些配置参数，如上图  
$.fn.dataTableExt.oPagination.listbox = {  
    /\*  
     \* Function: oPagination.listbox.fnInit  
     \* Purpose:  Initalise doom elements required for pagination with listbox input  
     \* Returns:  -  
     \* Inputs:   object:oSettings - data Tables settings object  
     \*             node:nPaging - the DIV which contains this pagination control  
     \*             function:fnCallbackDraw - draw function which must be called on update  
     \*/  
    "fnInit": function (oSettings, nPaging, fnCallbackDraw) {  
        var nInput = document.createElement('select');  
        var nPage = document.createElement('span');  
        var nOf = document.createElement('span');  
        nOf.className = "paginate\_of";  
        nPage.className = "paginate\_page";  
        if (oSettings.sTableId !== '') {  
            nPaging.setAttribute('id', oSettings.sTableId + '\_paginate');  
        }  
        nInput.style.display = "inline";  
        nPage.innerHTML = "Page ";  
        nPaging.appendChild(nPage);  
        nPaging.appendChild(nInput);  
        nPaging.appendChild(nOf);  
        $(nInput).change(function (e) { // Set DataTables page property and redraw the grid on listbox change event.  
            window.scroll(0,0); //scroll to top of page  
            if (this.value === "" || this.value.match(/[^0-9]/)) { /\* Nothing entered or non-numeric character \*/  
                return;  
            }  
            var iNewStart = oSettings.\_iDisplayLength \* (this.value - 1);  
            if (iNewStart > oSettings.fnRecordsDisplay()) { /\* Display overrun \*/  
                oSettings.\_iDisplayStart = (Math.ceil((oSettings.fnRecordsDisplay() - 1) / oSettings.\_iDisplayLength) - 1) \* oSettings.\_iDisplayLength;  
                fnCallbackDraw(oSettings);  
                return;  
            }  
            oSettings.\_iDisplayStart = iNewStart;  
            fnCallbackDraw(oSettings);  
        }); /\* Take the brutal approach to cancelling text selection \*/  
        $('span', nPaging).bind('mousedown', function () {  
            return false;  
        });  
        $('span', nPaging).bind('selectstart', function () {  
            return false;  
        });  
    },  
       
    /\*  
     \* Function: oPagination.listbox.fnUpdate  
     \* Purpose:  Update the listbox element  
     \* Returns:  -  
     \* Inputs:   object:oSettings - data Tables settings object  
     \*             function:fnCallbackDraw - draw function which must be called on update  
     \*/  
    "fnUpdate": function (oSettings, fnCallbackDraw) {  
        if (!oSettings.aanFeatures.p) {  
            return;  
        }  
        var iPages = Math.ceil((oSettings.fnRecordsDisplay()) / oSettings.\_iDisplayLength);  
        var iCurrentPage = Math.ceil(oSettings.\_iDisplayStart / oSettings.\_iDisplayLength) + 1; /\* Loop over each instance of the pager \*/  
        var an = oSettings.aanFeatures.p;  
        for (var i = 0, iLen = an.length; i < iLen; i++) {  
            var spans = an[i].getElementsByTagName('span');  
            var inputs = an[i].getElementsByTagName('select');  
            var elSel = inputs[0];  
            if(elSel.options.length != iPages) {  
                elSel.options.length = 0; //clear the listbox contents  
                for (var j = 0; j < iPages; j++) { //add the pages  
                    var oOption = document.createElement('option');  
                    oOption.text = j + 1;  
                    oOption.value = j + 1;  
                    try {  
                        elSel.add(oOption, null); // standards compliant; doesn't work in IE  
                    } catch (ex) {  
                        elSel.add(oOption); // IE only  
                    }  
                }  
                spans[1].innerHTML = " nbsp;of nbsp;" + iPages;  
            }  
          elSel.value = iCurrentPage;  
        }  
    }  
};**

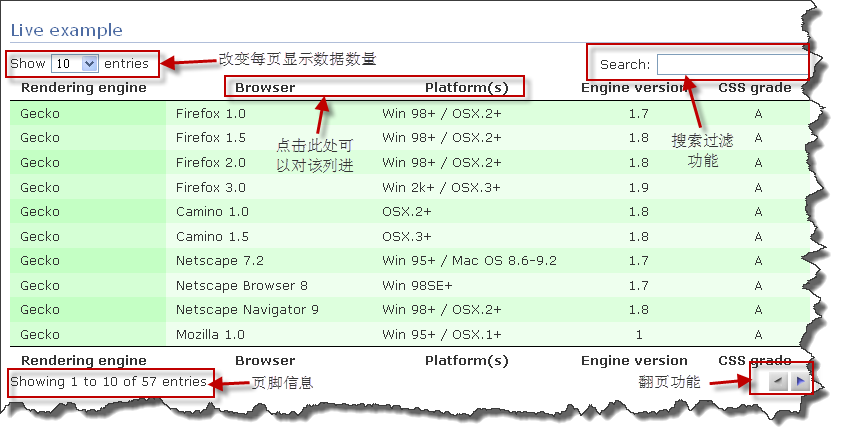
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| jQuery 的插件 data Tables 是一个优秀的表格插件，提供了针对表格的排序、浏览器分页、服务器分页、筛选、格式化等功能。data Tables 的网站上也提供了大量的演示和详细的文档进行说明，为了方便学习使用，这里一步一步进行说明。 首先，需要到 data Tables 的网站 <http://www.datatables.net/> 下载这个脚本库， 目前最新的版本是 1.8.2，下载的压缩包中使用的 jQuery 是 1.4.4 。现在 jQuery1.5.1 已经发布，所以，这里使用最新的 jQuery 1.5.1 。 然后，在网页中先加入 jQuery 的引用，然后，加入 data Tables 的引用。  引入CSS文件和JS文件 -------------------------------------------------------------------------- <style type="text/css" title="currentStyle">         @import "DataTables-1.8.1/media/css/demo\_page.css";         @import "DataTables-1.8.1/media/css/demo\_table.css";         @import "DataTables-1.8.1/media/css/demo\_table\_jui.css"; </style> <script type="text/javascript" language="javascript" src="DataTables-1.8.1/media/js/jquery.js"></script> <script type="text/javascript" language="javascript" src="DataTables-1.8.1/media/js/jquery.dataTables.js"></script>  --------------------------------------------------------------------------  -----------最简单的方式： $(document).ready(function() {  $("#example").dataTable(); });  ----------也可以自己定义各属性： <script type="text/javascript" language="javascript">         $(document).ready(function() {             $("#example").dataTable({ //                "bPaginate": true, //开关，是否显示分页器 //                "bInfo": true, //开关，是否显示表格的一些信息 //                "bFilter": true, //开关，是否启用客户端过滤器 //                "sDom": "<>lfrtip<>", //                "bAutoWith": false,//自动调整表格宽度 //                "bDeferRender": false, //                "bJQueryUI": false, //开关，是否启用JQueryUI风格 //                "bLengthChange": true, //开关，是否显示每页大小的下拉框 //                "bProcessing": true,//在初始化加载数据时，是否显示进度条 //                "bScrollInfinite": false,//是否显示滚动条 //                "sScrollY": "800px", //是否开启垂直滚动，以及指定滚动区域大小,可设值：'disabled','2000px' //                "bSort": true, //开关，是否启用各列具有按列排序的功能 //                "bSortClasses": true, //                "bStateSave": false, //开关，是否打开客户端状态记录功能。这个数据是记录在cookies中的， 打开了这个记录后，即使刷新一次页面，或重新打开浏览器，之前的状态都是保存下来的- ------当值为true时aoColumnDefs不能隐藏列 //                "sScrollX": "50%", //是否开启水平滚动，以及指定滚动区域大小,可设值：'disabled','2000%' //                "aaSorting": [[0, "asc"]], //                "aoColumnDefs": [{ "bVisible": false, "aTargets": [0]}]//隐藏列 //                "sDom": '<"H"if>t<"F"if>',                 "bAutoWidth": false, //自适应宽度                 "aaSorting": [[1, "asc"]],                 "sPaginationType": "full\_numbers",                 "oLanguage": {                     "sProcessing": "正在加载中......",                     "sLengthMenu": "每页显示 \_MENU\_ 条记录",                     "sZeroRecords": "对不起，查询不到相关数据！",                     "sEmptyTable": "表中无数据存在！",                     "sInfo": "当前显示 \_START\_ 到 \_END\_ 条，共 \_TOTAL\_ 条记录",                     "sInfoFiltered": "数据表中共为 \_MAX\_ 条记录",                     "sSearch": "搜索",                     "oPaginate": {                         "sFirst": "首页",                         "sPrevious": "上一页",                         "sNext": "下一页",                         "sLast": "末页"                     }                 } //多语言配置               });         });     </script>   ----------------页面：  对于 data Tables 来说，表格必须通过 thead 和 tbody 进行说明，如下所示，   1. <table cellpadding="0" cellspacing="0" border="0" class="display" id="example"> 2. <thead> 3. <tr> 4. <th> 5. Rendering engine 6. </th> 7. <th> 8. Browser 9. </th> 10. <th> 11. Platform(s) 12. </th> 13. <th> 14. Engine version 15. </th> 16. <th> 17. CSS grade 18. </th> 19. </tr> 20. </thead> 21. <tbody> 22. <tr class="odd gradeX"> 23. <td> 24. Trident 25. </td> 26. <td> 27. Internet Explorer 4.0 28. </td> 29. <td> 30. Win 95+ 31. </td> 32. <td class="center"> 33. 4 34. </td> 35. <td class="center"> 36. X 37. </td> 38. </tr>   复制代码  如果没有 thead 将会报错。 http://images.cnblogs.com/cnblogs_com/haogj/286292/o_no_THead_Error.png  最为简单的使用方式，就是零配置的方式。   1. /\* 2. \* Example init 3. \*/ 4. $(document).ready(function(){ 5. $('#example').dataTable(); 6. });   以下是在进行dataTable绑定处理时候可以附加的参数：  DataTable支持如下回调函数   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 回调函数名称 | 参数 | 返回值 | 默认 | 功能 | | fnCookieCallback | 1.string: Name of the cookie defined by DataTables 2.object: Data to be stored in the cookie 3.string: Cookie expires string 4.string: Path of the cookie to set | string: cookie formatted string (which should be encoded by using encodeURIComponent()) | null | 当每次cookies改变时，会触发这个函数调用 | | fnDrawCallback | 无 | 无 | 无 | 在每次table被draw完后调用，至于做什么就看着办吧 | | fnFooterCallback | 1.node : "TR" element for the footer 2.array array strings : Full table data (as derived from the original HTML) 3.int : Index for the current display starting point in the display array< 4.int : Index for the current display ending point in the display array 5.array int : Index array to translate the visual position to the full data array | 无 | 无 | 用于在每次重画的时候修改表格的脚部 | | fnFormatNumber | 1.int : number to be formatted | String : formatted string for DataTables to show the number | 有默认的 | 用于在大数字上，自动加入一些逗号，分隔开 | | fnHeaderCallback | 1.node : "TR" element for the header 2.array array strings : Full table data (as derived from the original HTML) 3.int : Index for the current display starting point in the display array 4.int : Index for the current display ending point in the display array 5.array int : Index array to translate the visual position to the full data array | 无 | 无 | 用于在每次draw发生时，修改table的header | | fnInfoCallback | 1.object: DataTables settings object 2.int: Starting position in data for the draw 3.int: End position in data for the draw 4.int: Total number of rows in the table (regardless of filtering) 5.int: Total number of rows in the data set, after filtering 6.string: The string that DataTables has formatted using it's own rules | string: The string to be displayed in the information element. | 无 | 用于传达table信息 | | fnInitComplete | 1.object:oSettings - DataTables settings object | 无 | 无 | 表格初始化完成后调用 | | fnPreDrawCallback | 1.object:oSettings - DataTables settings object | Boolean | 无 | 用于在开始绘制之前调用，返回false的话，会阻止draw事件发生；返回其它值，draw可以顺利执行 | | fnRowCallback | 1.node : "TR" element for the current row 2.array strings : Raw data array for this row (as derived from the original HTML) 3.int : The display index for the current table draw 4.int : The index of the data in the full list of rows (after filtering) | node : "TR" element for the current row | 无 | 当创建了行，但还未绘制到屏幕上的时候调用，通常用于改变行的class风格 | | fnServerData | 1.string: HTTP source to obtain the data from (i.e. sAjaxSource) 2.array objects: A key/value pair object containing the data to send to the server 3.function: Function to be called on completion of the data get process that will draw the data on the page. | void | $.getJSON | 用于替换默认发到服务端的请求操作 | | fnStateLoadCallback | 1.object:oSettings - DataTables settings object 2.object:oData - Object containing information retrieved from the state saving cookie which should be restored. For the exact properties please refer to the DataTables code. | Boolean - false if the state should not be loaded, true otherwise | 无 | 在cookies中的数据被加载前执行，可以方便地修改这些数据 | | fnStateSaveCallback | 1.object:oSettings - DataTables settings object 2.String:sValue - a JSON string (without the final closing brace) which should be stored in the state saving cookie. | String - the full string that should be used to save the state | 无 | 在状态数据被存储到cookies前执行，可以方便地做一些预操作 |   表格的效果 http://images.cnblogs.com/cnblogs_com/haogj/286292/o_zero.png 注意，红框表示了四个默认的设置效果，分别用来选择每页的行数，表格的过滤器，表格的信息和换页。 此时，使用了几个默认的参数设置。在 data Tables 中，参数名称的前缀用来说明参数的数据类型，很明显，b 表示布尔类型，i 表示整数类型，s 表示字符串类型。   * bPaginate: 是否分页，默认为 true，分页 * iDisplayLength : 每页的行数，每页默认数量:10 * sPaginationType: 分页样式，支持两种内置方式，two\_button 和 full\_numbers, 默认使用 two\_button。 * bLengthChange : 是否允许用户通过一个下拉列表来选择分页后每页的行数。行数为 10，25，50，100。这个设置需要 bPaginate 支持。默认为 true。 * bFilter: 启用或禁止数据过滤，默认为 true。 注意，如果使用过滤功能，但是希望关闭默认的过滤输入框，应使用 sDom * bInfo: 允许或者禁止表信息的显示，默认为 true，显示信息。   我们也可以通过传递一个初始化参数对象来改变这些设置。例如，下面的例子将每页的行数设置为 20 行。   1. $(function () { 2. $("#example").dataTable( 3. { 4. iDisplayLength: 20 5. } 6. ); 7. }); |

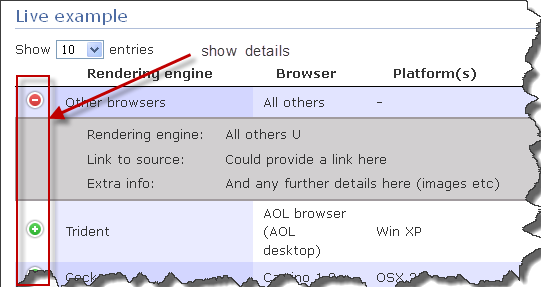
|  |  |  |
| --- | --- | --- |
| 属性名称 | 取值范围 | 解释 |
| bAutoWidth | true or false, default true | 是否自动计算表格各列宽度 |
| bDeferRender | true or false, default false | 用于渲染的一个参数 |
| bFilter | true or false, default true | 开关，是否启用客户端过滤功能 |
| bInfo | true or false, default true | 开关，是否显示表格的一些信息 |
| bJQueryUI | true or false, default false | 是否使用jquery ui themeroller的风格 |
| bLengthChange | true or false, default true | 开关，是否显示一个每页长度的选择条（需要分页器支持） |
| bPaginate | true or false, default true | 开关，是否显示（使用）分页器 |
| bProcessing | true or false, defualt false | 开关，以指定当正在处理数据的时候，是否显示“正在处理”这个提示信息 |
| bScrollInfinite | true or false, default false | 开关，以指定是否无限滚动（与sScrollY配合使用），在大数据量的时候很有用。当这个标志为true的时候，分页器就默认关闭 |
| bSort | true or false, default true | 开关，是否让各列具有按列排序功能 |
| bSortClasses | true or false, default true | 开关，指定当前列在排序时，是否增加classes 'sorting\_1', 'sorting\_2' and 'sorting\_3'，打开后，在处理大数据时，性能有所损失 |
| bStateSave | true or false, default false | 开关，是否打开客户端状态记录功能。这个数据是记录在cookies中的，打开了这个记录后，即使刷新一次页面，或重新打开浏览器，之前的状态都是保存下来的 |
| sScrollX | 'disabled' or  '100%' 类似的字符串 | 是否开启水平滚动，以及指定滚动区域大小 |
| sScrollY | 'disabled' or '200px' 类似的字符串 | 是否开启垂直滚动，以及­­指定滚动区域大小 |
| -- | -- | -- |
| 选项 |  |  |
| aaSorting | array array[int,string], 如[], [[0,'asc'], [0,'desc']] | 指定按多列数据排序的依据 |
| aaSortingFixed | 同上 | 同上。唯一不同点是不能被用户的自定义配置冲突 |
| aLengthMenu | default [10, 25, 50, 100]，可以为一维数组，也可为二维数组，比如：[[10, 25, 50, -1], [10, 25, 50, "All"]] | 这个为选择每页的条目数，当使用一个二维数组时，二维层面只能有两个元素，第一个为显示每页条目数的选项，第二个是关于这些选项的解释 |
| aoSearchCols | default null, 类似：[null, {"sSearch": "My filter"}, null,{"sSearch": "^[0-9]", "bEscapeRegex": false}] | 给每个列单独定义其初始化搜索列表特性（这一块还没搞懂） |
| asStripClasses | default ['odd', 'even'], 比如['strip1', 'strip2', 'strip3'] | 指定要被应用到各行的class风格，会自动循环 |
| bDestroy | true or false, default false | 用于当要在同一个元素上执行新的dataTable绑定时，将之前的那个数据对象清除掉，换以新的对象设置 |
| bRetrieve | true or false, default false | 用于指明当执行dataTable绑定时，是否返回DataTable对象 |
| bScrollCollapse | true or false, default false | 指定适当的时候缩起滚动视图 |
| bSortCellsTop | true or false, default false | （未知的东东） |
| iCookieDuration | 整数，默认7200，单位为秒 | 指定用于存储客户端信息到cookie中的时间长度，超过这个时间后，自动过期 |
| iDeferLoading | 整数，默认为null | 延迟加载，它的参数为要加载条目的数目，通常与bServerSide，sAjaxSource等配合使用 |
| iDisplayLength | 整数，默认为10 | 用于指定一屏显示的条数，需开启分页器 |
| iDisplayStart | 整数，默认为0 | 用于指定从哪一条数据开始显示到表格中去 |
| iScrollLoadGap | 整数，默认为100 | 用于指定当DataTable设置为滚动时，最多可以一屏显示多少条数据 |
| oSearch | 默认{ "sSearch": "", "bRegex": false, "bSmart": true } | 又是初始时指定搜索参数相关的，有点复杂，没搞懂目前 |
| sAjaxDataProp | 字符串，default 'aaData' | 指定当从服务端获取表格数据时，数据项使用的名字 |
| sAjaxSource | URL字符串，default null | 指定要从哪个URL获取数据 |
| sCookiePrefix | 字符串，default 'SpryMedia\_DataTables\_' | 当打开状态存储特性后，用于指定存储在cookies中的字符串的前缀名字 |
| sDom | default lfrtip (when bJQueryUI is false) or <"H"lfr>t<"F"ip> (when bJQueryUI is true) | 这是用于定义DataTable布局的一个强大的属性，另开专门文档来补充说明吧 |
| sPaginationType | 'full\_numbers' or 'two\_button', default 'two\_button' | 用于指定分页器风格 |
| sScrollXInner | string default 'disabled' | 又是水平滚动相关的，没搞懂啥意思 |

[jQuery dataTables 的使用方法.png](http://www.52share.net/attachment.php?aid=61363&k=5fc3fa60efab1a299ab24a9e91986a34&t=1357355012&fid=62&nothumb=yes) (13.42 KB)



jQuery的DataTables插件的使用方法

在做后台的时候并没有美工和前端工程师来配合你做页面，为了显示数据并有一定的美感，我们可以使用jQuery的DataTables插件来帮助我们完成任务  
1、DataTables的默认配置  
$(document).ready(function() {$(‘#example’).dataTable();} );  
示例：http://www.guoxk.com/html/DataTables/Zero-configuration.html  
2、DataTables的一些基础属性配置  
“bPaginate”: true, //翻页功能“bLengthChange”: true, //改变每页显示数据数量“bFilter”: true, //过滤功能“bSort”: false, //排序功能“bInfo”: true,//页脚信息“bAutoWidth”: true//自动宽度  
  
  
  
示例：http://www.guoxk.com/html/DataTables/Feature-enablement.html  
3、数据排序  
$(document).ready(function() {$(‘#example’).dataTable( {“aaSorting”: [[ 4, "desc" ]]} );} );  
从第0列开始，以第4列倒序排列  
示例：http://www.guoxk.com/html/DataTables/Sorting-data.html  
4、多列排序  
示例：http://www.guoxk.com/html/DataTables/Multi-column-sorting.html  
5、隐藏某些列  
          $(document).ready(function() {$(‘#example’).dataTable( {“aoColumnDefs”: [{ "bSearchable": false, "bVisible": false, "aTargets": [ 2 ] },{ “bVisible”: false, “aTargets”: [ 3 ] }] } );} );  
示例：http://www.guoxk.com/html/DataTables/Hidden-columns.html  
6、改变页面上元素的位置  
$(document).ready(function() {$(‘#example’).dataTable( {“sDom”: ‘<”top”fli>rt<”bottom”p><”clear”>’} );} );//l- 每页显示数量//f – 过滤输入//t – 表单Table//i – 信息//p – 翻页//r – pRocessing//< and > – div elements//<”class” and > – div with a class//Examples: <”wrapper”flipt>, <lf<t>ip>  
示例：http://www.guoxk.com/html/DataTables/DOM-positioning.html  
7、状态保存，使用了翻页或者改变了每页显示数据数量，会保存在cookie中，下回访问时会显示上一次关闭页面时的内容。  
$(document).ready(function() {$(‘#example’).dataTable( {“bStateSave”: true} );} );  
示例：http://www.guoxk.com/html/DataTables/State-saving.html  
8、显示数字的翻页样式  
$(document).ready(function() {$(‘#example’).dataTable( {“sPaginationType”: “full\_numbers”} );} );  
示例：http://www.guoxk.com/html/DataTables/Alternative-pagination-styles.html  
9、水平限制宽度  
$(document).ready(function() {$(‘#example’).dataTable( {“sScrollX”: “100%”,“sScrollXInner”: “110%”,“bScrollCollapse”: true} );} );  
示例：http://www.guoxk.com/html/DataTables/Horizontal.html  
10、垂直限制高度  
示例：http://www.guoxk.com/html/DataTables/Vertical.html  
11、水平和垂直两个方向共同限制  
示例：http://www.guoxk.com/html/DataTables/HorizontalVerticalBoth.html  
12、改变语言  
$(document).ready(function() {$(‘#example’).dataTable( {“oLanguage”: {“sLengthMenu”: “每页显示 \_MENU\_ 条记录”,“sZeroRecords”: “抱歉， 没有找到”,“sInfo”: “从 \_START\_ 到 \_END\_ /共 \_TOTAL\_ 条数据”,“sInfoEmpty”: “没有数据”,“sInfoFiltered”: “(从 \_MAX\_ 条数据中检索)”,“oPaginate”: {“sFirst”: “首页”,“sPrevious”: “前一页”,“sNext”: “后一页”,“sLast”: “尾页”},“sZeroRecords”: “没有检索到数据”,“sProcessing”: “<img src=’./loading.gif’ />”}} );} );  
示例：http://www.guoxk.com/html/DataTables/Change-language-information.html  
13、click事件  
示例：http://www.guoxk.com/html/DataTables/event-click.html  
14/配合使用tooltip插件  
示例：http://www.guoxk.com/html/DataTables/tooltip.html  
15、定义每页显示数据数量  
$(document).ready(function() {$(‘#example’).dataTable( {“aLengthMenu”: [[10, 25, 50, -1], [10, 25, 50, "All"]]} );} );  
示例：http://www.guoxk.com/html/DataTables/length\_menu.html  
16、row callback  
示例：http://www.guoxk.com/html/DataTables/RowCallback.html  
最后一列的值如果为“A”则加粗显示  
17、排序控制  
$(document).ready(function() {$(‘#example’).dataTable( {“aoColumns”: [null,{ "asSorting": [ "asc" ] },{ “asSorting”: [ "desc", "asc", "asc" ] },{ “asSorting”: [ ] },{ “asSorting”: [ ] }]} );} );  
示例：http://www.guoxk.com/html/DataTables/sort.html说明：第一列点击按默认情况排序，第二列点击已顺序排列，第三列点击一次倒序，二三次顺序，第四五列点击不实现排序  
18、从配置文件中读取语言包  
$(document).ready(function() {$(‘#example’).dataTable( {“oLanguage”: {“sUrl”: “cn.txt”}} );} );  
19、是用ajax源  
$(document).ready(function() {$(‘#example’).dataTable( {“bProcessing”: true,“sAjaxSource”: ‘./arrays.txt’} );} );  
示例：http://www.guoxk.com/html/DataTables/ajax.html  
20、使用ajax，在服务器端整理数据  
$(document).ready(function() {$(‘#example’).dataTable( {“bProcessing”: true,“bServerSide”: true,“sPaginationType”: “full\_numbers”,  
“sAjaxSource”: “./server\_processing.php”,/\*如果加上下面这段内容，则使用post方式传递数据“fnServerData”: function ( sSource, aoData, fnCallback ) {$.ajax( {“dataType”: ‘json’,“type”: “POST”,“url”: sSource,“data”: aoData,“success”: fnCallback} );}\*/“oLanguage”: {“sUrl”: “cn.txt”},“aoColumns”: [{ "sName": "platform" },{ "sName": "version" },{ "sName": "engine" },{ "sName": "browser" },{ "sName": "grade" }]//$\_GET['sColumns']将接收到aoColumns传递数据} );} );  
示例：http://www.guoxk.com/html/DataTables/ajax-serverSide.html  
21、为每行加载id和class  
服务器端返回数据的格式：  
{“DT\_RowId”: “row\_8″,“DT\_RowClass”: “gradeA”,“0″: “Gecko”,“1″: “Firefox 1.5″,“2″: “Win 98+ / OSX.2+”,“3″: “1.8″,“4″: “A”},  
示例：http://www.guoxk.com/html/DataTables/add\_id\_class.html  
22、为每行显示细节，点击行开头的“+”号展开细节显示  
示例：http://www.guoxk.com/html/DataTables/with-row-information.html



23、选择多行  
示例：http://www.guoxk.com/html/DataTables/selectRows.html  
22、集成jQuery插件jEditable  
示例：http://www.guoxk.com/html/DataTables/jEditable-integration.html  
示例打包下载：http://www.guoxk.com/html/DataTables/DataTables.rar