

EasyDataTable

AJAX 分页插件使用手册

使用 Ajax 分页可以提高数据加载和显示速度，减少网络流量，提升客户体验度；同时能够只刷新局部，解决当页面上有多个数据显示表格区域时，传统的分页方式会导致页面全部刷新。

EasyDataTable AJAX 分页插件是基于 jQuery 最好的纯 Ajax 分页插件，简单、易用、灵活；自带分页标签，支持自定义分页；支持排序；内置 EasyDataTable 表达式语言和事件支持能够通过 JavaScript 编程增强分页。

兼容性：EasyDataTable 完全兼容 IE6 及以上版本、Firefox、Chrome、Safari、Opera 等各内核（Trident、Gecko、Webkit、Presto）浏览器，并兼容多平台及系统（PC，TabletPC，Mobile）。

与 Ext 分页比较 EasyDataTable 的特点：

简单：EasyDataTable 需要更少的资源加载，非常轻量级

易用：几乎无需 JS 代码，全面封装，JS 零编程即可实现 Ajax 分页功能

灵活：更少的限制，自带表达式语言，事件支持，从 UI 到功能均可自定义和扩展

目录：

1. 在页面引入 EasyDataTable 核心的 js 和 css 文件	3
2. 对 Ajax 分页表格进行数据初始化	3
2.1 方法一：html，使用 easydatatable 类样式	3

2.2 方法二：Javascript，使用 DataTable.load() 函数	3
2.2.1 EasyDataTable 初始化参数	4
3. 分页表格结构	8
4. EasyDataTable Expression Language——表达式语言使用	11
4.1 EasyDataTable Property Expression 属性表达式 {数据属性}	11
4.2 EasyDataTable Statement Expression 语句表达式 %{表达式语句}%	11
5. EasyTableData 内置数据属性	12
6. 排序支持	13
7. 服务器端数据要求	14
8. 刷新指定数据表格	15
9. 默认分页条数设置	15
10. 分页主题设置	15
9.1 使用 JavaScript 指定分页主题	16
9.2 在分页 DIV 标签上指定分页主题	16
9.3 取消分页和主题	16
11. 自定义分页	16
12. 表格 AJAX 分页实例	17
12.1 指定分页主题（DataTable.SIMPLE_PAGE），加入 Loading 提示	17
12.2 判断语句 DataTable 表达式使用，默认 DataTable.SIMPLE_FULL 分页主题 带复选框和自动编号	19
12.3 带搜索条件分页	22
12.3.1 搜索按钮提交搜索数据的两种方法	22
12.3.2 实例	22

12.4 带 start 和 end 数据加载事件处理函数的分页	25
13. EasyDataTable 分页标签国际化支持	27
13.1 默认分页标签文字和语言	27
13.2 自定义分页标签文字和语言	28

快速开发步骤：

1. 在页面引入 EasyDataTable 核心的 js 和 css 文件

```
<link rel="stylesheet" href="css/datatables.css" type="text/css" />
<script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
<script type="text/javascript" src="js/easy.datatables.js"></script>
```

2. 对 Ajax 分页表格进行数据初始化

2.1 方法一：html，使用 easydatatable 类样式

给表格添加 easydatatable 类样式进行初始化

```
<table class="datatable easydatatable" id="datatable3" width="760px" align="center">
```

2.2 方法二：Javascript，使用 DataTable.load() 函数

使用 DataTable.load(表格 id，参数信息) 函数对要进行 Ajax 分页的表格进行数据初始化

DataTable.load(tableid [, easydataParameters]);

参数说明：

tableid: 必须参数, 显示数据的表格 id

easydataParameters : 可选参数, 指定 EasyDataTable 的参数信息，支持的参数：

```

{
    pagetheme: '分页主题',
    loading: '是否显示加载提示',
    language: '分页标签的语言',
    start: '数据开始加载事件处理函数',
    end: '数据结束加载事件处理函数',
    row: '初始每页加载显示数量'
}

```

2.2.1 EasyDataTable 初始化参数

- **pagetheme**——可选,分页标签主题，支持两套可选分页主题：

DataTable.FULL_PAGE (值为 FULL,完全主题，默认显示所有分页选项，**默认值**)

每页显示 5 条，共 54 条 共 11 页 首页 上一页 1 2 3 4 5 6 7 下一页 末页 1 >>

DataTable.SIMPLE_PAGE(值为 SIMPLE,简单主题，不显示当前页前后页快速跳转标签)

每页显示 5 条，共 54 条 共 11 页 首页 上一页 下一页 末页 1 >>

NO(取消主题，使用自定义分页)

参考第 10 章节《自定义分页》。

支持HTML增强：pagetheme 参数也可通分页 div 的 pagetheme 属性设置 加载顺序为 html、javascript，后加载的参数会覆盖前面的值。参考第 9 章节《分页主题设置》。

- **loading**——可选,分页加载数据时表格的显示方式，可选值为 **default**、**show**、**none**、**hide** 或任意值。

default: 默认分页加载方式，分页加载时禁用表格操作(禁用超链，按钮)，表格数据显示为灰色

示例：

<input type="checkbox"/>	count	id	name ↑↓	info	操作
<input type="checkbox"/>	3==11=== == ==	No.10	USER_10	INFO_10	查看 修改 删除
<input type="checkbox"/>	3==11=== == ==	No.11	USER_11	INFO_11	查看 修改 删除

show: 显示加载提示方式，分页加载时显示 DataTable.LOADING_MSG 定义的 loading 提示内容（可修改）"数据正在读取中....."

示例：

<input type="checkbox"/>	count	id	name ↑↓	info	操作
数据正在读取中.....					

none: 隐藏数据展示行方式，分页加载时隐藏数据展示行的数据+单元格，显示完全为空白

示例：

<input type="checkbox"/>	count	id	name ↑↓	info	操作

hide: 隐藏数据内容方式，分页加载时仅隐藏数据展示行的数据，保留显示单元格边框

示例：

<input type="checkbox"/>	count	id	name ↑↓	info	操作

任意值： 直接将该内容作为分页加载时提示内容（等同 show 方式），支持 HTML 内容

示例：

loading: "<div>
数据正在加载中.....</div>"

<input type="checkbox"/>	count	id	name ↑↓	info	操作
 数据正在加载中.....					

支持 HTML 增强：loading 参数也可通过数据表格的 loading 属性设置，加载顺序为 html、javascript，后加载的参数会覆盖前面的值。

- **language**——可选,设置分页标签显示的语言。

默认分页配置定义在DataTable.MSG属性中，默认值为：

```
DataTable.MSG={  
    first:"首页",  
    previous:"上一页",  
    next:"下一页",  
    last:"末页",  
    totalCount:"共{0}条",  
    totalPage:"共{0}页",  
    rowPerPage:"每页显示{0}条"  
}
```

可直接修改和重新定义该属性作为默认分页文字和语言。

也可根据需要在初始化参数中为数据表格重新定义,{0}为显示相应数据的占位符,必须存在。

```
<script type="text/javascript">  
$(function(){  
    var pageLanguage={  
        "first":'first',  
        "previous":'previous',  
        "next":'next',  
        "last":'last',  
        "totalPage":'total {0} pages',  
        "totalCount":'total {0} rows',
```

```

        "rowPerPage": 'page for {0} rows',

    };

    DataTable.load("datatable",{
        "pagetheme": DataTable.SIMPLE_PAGE,
        "loading": true,
        "language": pageLanguage
    });

    DataTable.load("datatable2");

    DataTable.load("datatable3");

    DataTable.load("datatable4");

});

</script>

```

- **start**——可选,设置每一次数据加载开始时的处理函数

```

/*
o: 当前表格对象

initFlag: true代表第一次加载数据（初始化表格），false代表分页加载
*/
"start": function(o, initFlag) {
    if (initFlag) { //第一次加载（未初始化）
        console.info('init start...');
    } else {
        console.info('load start...');
    }
}
}

```

- **end**——可选,设置每一次数据加载结束时的处理函数

```
/*
o: 当前表格对象

initFlag: true代表第一次加载数据（初始化表格），false代表分页加载
*/
"end":function(o,initFlag){

    if(initFlag){ //第一次加载（未初始化）

        console.info('init end...');
    }else{
        console.info('load end...');
    }

}
```

- **row**——可选,设置初始分页加载的条数

在初始化参数中指定默认分页加载的数据条数。

如不设置该参数，将使用 `DataTable.DEFAULT_ROW` 属性的值默认值 5（可修改）。

支持 HTML 增强：row 参数也可通过分页 DIV 的 row 属性设置 加载顺序为 html,javascript , 后加载的参数会覆盖前面的值。

3. 分页表格结构

```
<form action="服务器分页地址">
```

```
<!-- 数据展示表格 -->
```

```
<table id="表格 id">
```

```
<!-- 表头行 -->
```

```
<tr><th></th> ..... </tr>
```

```
<!-- 数据展示行 -->
```



```

        <tr style="display: none;"> <td> </td>      ..... </tr>

</table>

<!-- 分页行 -->

<div class="panelBar" style="width: 760px;" size="5,10,30,50">    </div>

</form>

```

示例：

```

<!-- 创建表单 action值为分页处理的请求地址 -->

<form action="doPage.jsp" name="myform">

    <div style="height: 260px;overflow:auto;width: 780px;">

        <!--显示数据的表格，必须指定id，EasyDatatable通过id初始化分页数据-->

        <table class="datatable" id="datatable" width="760px" align="center">

            <!--表头行-->

            <tr>

                <th width="40">

                    <!-- 调用DataTable.checkAll(this,'复选框名称')，可实现复选框全选/全不选功能 -->

                    <input type="checkbox" onclick="DataTable.checkAll(this,'mychk'" />

                </th>

                <th width="80">数量</th>

                <th width="100">编号</th>

                <th width="100">姓名</th>

```

```
<th width="100">信息</th>
```

```
<th>操作</th>
```

```
</tr>
```

```
<!-- 数据展示行，使用 {属性} 属性表达式获取数据 -->
```

```
<tr style="display: none;">
```

```
<td style="text-align:center,height: 45px;">
```

```
<!--自定义复选框,value为对象的id-->
```

```
<input type="checkbox" name="mychk" value="{id}"/>
```

```
</td>
```

```
<!--使用内置的datatableCount属性，显示数据条数-->
```

```
<td align="center" style="text-align:center,height: 45px;">{datatableCount}</td>
```

```
<td style="text-align:center;color:#00f">No.{id}</td>
```

```
<td align="center">{name}</td>
```

```
<td>{info}</td>
```

```
<td align="center" style="width: 120px">
```

```
<a href="doUser.jsp?o=show&id={id}">查看</a>
```

```
<a href="doUser.jsp?o=edit&id={id}">修改</a>
```

```
<a href="doUser.jsp?o=delete&id={id}">删除</a>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
</div>
```

```
<!--分页标签部分，使用size属性设置每页显示条数下拉菜单可选值，使用,分隔-->
```

```
<div class= "panelBar" style="width: 760px;" size="5,10,30,50">

</div>

</form>
```

4. EasyDataTable Expression Language——表达式语言使用

4.1 EasyDataTable Property Expression 属性表达式 *{数据属性}*

属性表达式用于在数据展示行，显示指定的属性值。

在属性表达式中可以直接引用数据属性来获得指定属性的数据，并支持各种数学、比较等 JavaScript 基本运算符进行运算。

`{id}` `{name}`

4.2 EasyDataTable Statement Expression 语句表达式 *%{表达式语句}%*

语句表达式用于在数据展示行，使用编程语句进行控制编程，在语句表达式中支持使用 JavaScript 编写表达式代码；支持直接调用数据属性；也支持使用 EasyDataTable 属性表达式。

语句表达式执行的结果必须使用 EasyDataTable 语句表达式标准输出方法输出：

`DataTable.out("内容");`

<!-- 支持JavaScript语言编写表达式，支持使用 属性名（以变量方式引用和处理） 或

EasyDataTable 属性表达式（必须使用引号定义在字符串中） 直接引用属性值 --%>

%{

```
var res=name+" {name}";
```

```
DataTable.out(res);
```

}%

%{

```
if(id%2==0){
```

```
var op='<a href= "doUser.jsp?o=show&id='+id+'">查看</a>&nbsp;&nbsp;&nbsp;';
```

```
op+='<a href= "doUser.jsp?o=edit&id={id }">修改</a>';
```

```
DataTable.out(op);
```

```
}else{
```

```
DataTable.out('<a href= "doUser.jsp?o=show&id={id }">查看
```

```
</a>&nbsp;&nbsp;&nbsp;<a href= "doUser.jsp?o=edit&id={id }">修改</a>&nbsp;&nbsp;&nbsp;<a
```

```
href= "doUser.jsp?o=delete&id={id }">删除</a>');
```

```
}
```

}%

5. EasyTableData 内置数据属性

仅在数据展示行有效：

datatableIndex:可获得数据在当前页的索引

datatableCount:可获得数据在当前页的个数

key:Map 数据集合时可用来获取数据对于的键

数据展示行+分页标签均可使用：

pageNo:当前页

maxPage:总页数

rowPerPage:每页显示条数

totalCount:数据总条数

order:排序字段

sort:排序方式，desc 或 asc

例如：

当前数据在所有数据中的索引： $\{datatableIndex+(pageNo-1)*rowPerPage\}$

当前数据在所有数据中的个数： $\{datatableCount+(pageNo-1)*rowPerPage\}$

6. 排序支持

在**表头行**需要排序字段对应的单元格上添加 **sort=" 排序字段名称"** 属性即可。

```
<tr>

    <th width= "80">count</th>

    <th width= "80">index</th>

    <th width= "80">{index+1}</th>

    <th width= "100" order= "id">id</th>

    <th width= "100" order= "name">name</th>

    <th width= "100" order= "info">info</th>

    <th >操作</th>

</tr>
```

id ↑↓

name ↑↓

info ↑↓

操作

可排序列会显示排序箭头。点击即可发送排序字段 order 和排序方式 sort , 实现升序降序切换。

| id ↑ | name ↑↓ | info ↑↓ | 操作 |
|------|---------|---------|----|
|------|---------|---------|----|

服务器端通过 order 和 sort 参数名获取排序信息。

```
String sort = request.getParameter("sort");
```

```
String order = request.getParameter("order");
```

刷新表格：

使用 DataTable.reload('tableid');可以刷新指定数据表格，恢复到没有排序状态。

```
<div onclick="DataTable.reload('datatable3')">刷新</div>
```

7. 服务器端数据要求

● 服务器端必须输出四条如下名称的分页参数 JSON 结果：

data:数据集，支持 List 和 Map 集合(内置属性 key 可获取 Map 的键 ,Map 的值无需使用 value.
前缀)

pageNo:当前第几页，数字

rowPerPage:每页显示条数，数字

totalCount:数据总条数，数字

可选参数：

[**order**]:排序字段

[**sort**]:排序方式，desc 或 asc

● 如果服务器端分页参数封装在 PageBean 中，例如：

```
public class PageBean {  
  
    private List data;  
  
    private int pageNo;  
  
    private int rowPerPage;
```

```

    private int totalCount;

    //setters&getters .....
}

```

返回的是 PageBean 对象，则在 table 上加入 value 属性，指定服务器端输出的包含分页参数的分页对象的 json 对象名称：

```

<table class="datatable easydatatable" id="datatable3" width="760px" align="center"
value="pb">

```

8. 刷新指定数据表格

```

DataTable.reload( "tableId" ); //取消排序效果，刷新表格，重新加载数据

```

9. 默认分页条数设置

在初始化分页参数中使用 row 参数可以指定默认的分页条数。如果不设置该参数，将使用

DataTable.DEFAULT_ROW 属性的值默认值 5（可修改）。

row 属性同事支持使用 HTML 增强：

```

<div class="panelBar" style="width: 760px;" size="5,10,30,50"
row="10">

</div>

```

10. 分页主题设置

DataTable 内置分页实现，并提供了两套主题：

DataTable.FULL_PAGE（完全主题，默认显示所有分页选项，默认值）

每页显示 5 条，共 54 条 共 11 页 首页 上一页 1 2 3 4 5 6 7 下一页 末页 1 >>

DataTable.SIMPLE_PAGE(简单主题，不显示当前页前后页快速跳转标签)

每页显示 5 条，共 54 条

共 11 页 首页 上一页 下一页 末页 1 >>

9.1 使用 JavaScript 指定分页主题

```
DataTable.load("datatable",{  
    "pagetheme":DataTable.SIMPLE_PAGE  
});
```

9.2 在分页 DIV 标签上指定分页主题

```
<div class="panelBar" style="width: 760px;height: 40px;" size="5,10,30,50"  
pagetheme="FULL">
```

```
<div class="panelBar" style="width: 760px;height: 40px;" size="5,10,30,50"  
pagetheme="SIMPLE">
```

9.3 取消分页和主题

使用 display:none 隐藏，或直接删除分页标签部分即可。

```
<div class="panelBar" style="width: 760px;height: 40px;display: none;" size="5,10,30,50" >
```

11.自定义分页

添加 **pagetheme=" no"** 属性（或通过 EasyDataTable 初始化参数设置），调用 **DataTable.go('加载数据的表格id' ; 页数' ,[每页显示条数])** 函数，即可实现自定义分页跳转；也可使用 **<input type="hidden" name="rowPerPage" value="8"/>** 隐藏域指定默认每页显示条数。


```
<div class="panelBar" style="width: 760px,height: 40px;" size="5,10,30,50"
```

```
pagetheme="no">
```

```
<input type="hidden" name="rowPerPage" value="8"/>
```

当前第{pageNo}页/共{maxPage}页 每页{rowPerPage}条/共{totalCount}条

```
<input type="button" value="首页" onclick="DataTable.go('datatable7',1)"/>
```

```
<input type="button" value="上一页" onclick="DataTable.go('datatable7', '{pageNo-1}')"/>
```

```
<input type="button" value="下一页" onclick="DataTable.go('datatable7',
```

```
'{pageNo+1}')"/>
```

```
<input type="button" value="末页" onclick="DataTable.go('datatable7', '{maxPage}')"/>
```

```
</div>
```

在函数调用时参数如果使用了 EasyDataTable 表达式 (如页数{pageNo-1}) 则需要通过单引号引起来, 作为一个字符串参数, 如 `onclick="DataTable.go('datatable7', '{pageNo-1}')"/>` 中的 `'{pageNo-1}'`。

12. 表格 AJAX 分页实例

12.1 指定分页主题 (DataTable.SIMPLE_PAGE), 加入 Loading 提示

```
<script type="text/javascript">
```

```
$(function(){
```

```
    DataTable.load("datatable", DataTable.SIMPLE_PAGE, true);
```

```
});
```

```
</script>
```

```
<div style="margin: 20 0 10 0; font-size: 28px;">
```

DataTable.SIMPLE_PAGE 分页主题 带Loading提示

```
</div>
```

```
<form action="doPage.jsp" name="myform">
```

```
<div style="height: 260px;overflow:auto;width: 780px;">
```

```
<table class="datatable" id="datatable" width="760px" align="center">
```

```
<tr>
```

```
<th width="40">
```

```
<input type="checkbox"
```

```
onclick="DataTable.checkAll(this,'mychk')"/> <!-- CheckAll -->
```

```
</th>
```

```
<!-- datatableCount -->
```

```
<th width="80">count</th>
```

```
<th width="100">id</th>
```

```
<th width="100">name</th>
```

```
<th width="100">info</th>
```

```
<th >操作</th>
```

```
</tr>
```

```
<!-- 数据展示行 -->
```

```
<tr style="display: none;">
```

```
<td style="text-align:center;height: 45px;">
```

```
<input type="checkbox" name="mychk" value="{id}"/>
```

```

        </td>

        <td align = "center" style="text-align:center,height:
45px;">{datatableCount }</td>

        <td style="text-align:center;color:#00f">No.{id}</td>

        <td align = "center">{name}</td>

        <td>{info}</td>

        <td align = "center" style="width: 120px">

            <a href = "doUser.jsp?o=show&id={id }">查看</a>

            <a href = "doUser.jsp?o=edit&id={id }">修改</a>

            <a href = "doUser.jsp?o=delete&id={id }">删除</a>

        </td>

    </tr>

</table>

</div>

<div class = "panelBar" style="width: 760px;" size = "5,10,30,50">

</div>

</form>

```

12.2 判断语句 DataTable 表达式使用 , 默认 DataTable.SIMPLE_FULL 分页主题 带复选框和自动编号

```

<div style="margin: 40 0 10 0; font-size: 28px;">判断语句DataTable表达式使用</div>

<form action = "doPage.jsp" name = "myform">

<div style="height: 260px;overflow:auto;width: 780px;">

```

```

<table class="datatable easydatatable" id="datatable3" width="760px"
align="center">

    <tr>

        <!-- checkbox -->

            <th width="40">

                <input type="checkbox"

onclick="DataTable.checkAll(this,'mychk')" /> <!-- CheckAll -->

            </th>

            <!-- datatableIndex,datatableCount -->

            <th width="80">count</th>

            <th width="80">index</th>

            <th width="80">{index+1}</th>

            <th width="100">id</th>

            <th width="100">name</th>

            <th width="100">info</th>

            <th >操作</th>

        </tr>

        <!-- 数据展示行 -->

        <tr style="display: none;">

            <td style="text-align:center,height: 45px;">

                <input type="checkbox" name="mychk" value="{id }"/>

            </td>

            <td align="center"> {datatableCount+(pageNo-1)*rowPerPage}</td>

            <td align="center"> {datatableIndex+(pageNo-1)*rowPerPage}</td>

```

```

1)*rowPerPage+1}</td>

<td style="text-align:center;color:#00f">No.{id}</td>

<td align="center">{name}</td>

<td>{info}</td>

<td align="center">

<!-- DataTable 表达式 -->

%{

    if(id%2==0){

        var op='<a href="doUser.jsp?o=show&id='+id+'">查看

</a> &nbsp;&nbsp;&nbsp;';

        op+='<a href="doUser.jsp?o=edit&id={id}">修改</a>';

        DataTable.out(op);

    }else{

        DataTable.out('<a href="doUser.jsp?o=show&id={id}">查看

</a> &nbsp;&nbsp;&nbsp;<a href="doUser.jsp?o=edit&id={id}">修改</a> &nbsp;&nbsp;&nbsp;<a

href="doUser.jsp?o=delete&id={id}">删除</a>');

    }

}%

</td>

</tr>

</table>

```

```
</div>
```

```
<div class="panelBar" style="width: 760px;" size="5,10,30,50">
```

```
</div>
```

```
</form>
```

12.3 带搜索条件分页

12.3.1 搜索按钮提交搜索数据的两种方法

方法一：给搜索按钮添加 `onclick="DataTable.load('当前数据表格 id')"`

```
<input type="button" class="btn_test" onclick="DataTable.load('datatable4')"  
value="立即查询"/>
```

方法二：给搜索按钮直接加 `easydatatable_search` 类样式

```
<input type="button" class="btn_test easydatatable_search" value="立即查询"/>
```

12.3.2 实例

```
<div style="margin: 40 0 10 0; font-size: 28px;">带搜索条件分页</div>
```

```
<form action="doPage.jsp" name="myform">
```

```
<!--搜索条件部分，通过搜索按钮查询-->
```

```
<div style="margin: 20px 0px;">
```

```
用户名：<input type="text" name="user.name" class="txt_test"/>
```

信息 : `<input type="text" name="user.info" class="txt_test"/>`

`<input type="button" class="btn_test"`

`onclick="DataTable.load('datatable4')" value="立即查询"/>`

`</div>`

`<div style="height: 260px;overflow:auto;width: 780px;">`

`<table class="datatable easydatatable" id="datatable4" width="760px"`
`align="center">`

`<tr>`

`<!-- checkbox -->`

`<th width="40">`

`<input type="checkbox"`

`onclick="DataTable.checkAll(this,'mychk')" /> <!-- CheckAll -->`

`</th>`

`<!-- datatableIndex,datatableCount -->`

`<th width="80">count</th>`

`<th width="80">index</th>`

`<th width="80">{index+1}</th>`

`<th width="100">id</th>`

`<th width="100">name</th>`

`<th width="100">info</th>`

`<th >操作</th>`

`</tr>`

```
<!-- 数据展示行 -->
```

```
<tr style="display: none;">
```

```
<td style="text-align:center;height: 45px;">
```

```
<input type="checkbox" name="mychk" value="{id }"/>
```

```
</td>
```

```
<td align="center">{datatableCount+(pageNo-1)*rowPerPage}</td>
```

```
<td align="center">{datatableIndex+(pageNo-1)*rowPerPage}</td>
```

```
<td align="center">{datatableIndex+(pageNo-
```

```
1)*rowPerPage+1}</td>
```

```
<td style="text-align:center;color:#00f">No.{id}</td>
```

```
<td align="center">{name}</td>
```

```
<td>{info}</td>
```

```
<td align="center">
```

```
<!-- DataTable 表达式 -->
```

```
%{
```

```
if(id%2==0){
```

```
    DataTable.out('<a href="doUser.jsp?o=show&id={id}"
```

```
target="ajax">查看</a>&nbsp;&nbsp;&nbsp;<a href="doUser.jsp?o=edit&id={id}"
```

```
target="ajax">修改</a>');
```

```
    }else{
```

```
        DataTable.out('<a href="doUser.jsp?o=show&id={id}"
```

```
target="ajax">查看</a>&nbsp;&nbsp;&nbsp;<a href="doUser.jsp?o=edit&id={id}"
```

```
target="ajax">修改</a>&nbsp;&nbsp;&nbsp;<a href="doUser.jsp?o=delete&id={id}"
```

```
target="ajax">删除</a>');
```



```

        }

        }%

    </td>

</tr>

</table>

</div>

<div class= "panelBar" style="width: 760px;" size= "5,10,30,50">

</div>

</form>

```

12.4 带 start 和 end 数据加载事件处理函数的分页

```

<script type="text/javascript">
    $(function() {
        //实现初始化loading效果，加载完成后隐藏loading，显示初始数据
        DataTable.load("datatable_event",{
            "start":function(dataTableObj,initFlag){
                if(initFlag){ //第一次加载，未初始化
                    console.info('init start...');
                    $("#loading").show(); //显示loading提示DIV
                    $("#dataDiv").hide(); //隐藏数据div
                    $("#dataPageDiv").hide(); //隐藏分页div
                }else{
                    console.info('load start...');
                }
            },
            "end":function(dataTableObj,initFlag){

```

```

        if (initFlag) { //第一次加载，未初始化
            console.info('init end...');

            $("#loading").hide(); //隐藏loading提示DIV

            $("#dataDiv").show(); //显示数据div

            $("#dataPageDiv").show(); //显示分页div
        } else {
            console.info('load end...');
        }
    }

    });
</script>

```

```

<form action="doPage_slow.jsp" name="myform">
    <!-- loading提示DIV,第一次加载数据时显示 -->

    <div id="loading" style="border:1px solid #efefef; text-align:
center; width: 780px; height: 285px; display: none; font-size: 14px;">

        <br/>数据正在加载中.....

    </div>

    <div style="height: 260px; overflow:auto; width: 780px;" id="dataDiv">
        <table class="datatable" id="datatable_event" width="760px"
align="center" value="pb">
            <tr>
                <th width="40">
                    <input type="checkbox"
onclick="DataTable.checkAll(this, 'mychk')" /> <!-- CheckAll -->
                </th>
                <!-- datatableCount -->
                <th width="80">count</th>
                <th width="100">id</th>
                <th width="150" order="name">name</th>
                <th width="150">info</th>
                <th >operation</th>
            </tr>
            <!-- Data Show Row-->
            <tr style="display: none;">
                <td style="text-align:center; height: 45px;">
                    <input type="checkbox" name="mychk" value="{id }"/>
                </td>
                <td align="center" style="text-align:center; height:

```

```

45px; ">
        {other} ==
        {sort }==={order}</td>
        <td style="text-align:center;color:#00f">No.{id}</td>
        <td align="center">{name}</td>
        <td>{info}</td>
        <td align="center" >
            <a href="doUser.jsp?o=show&id={id }">show</a>
            <a href="doUser.jsp?o=edit&id={id }">edit</a>
            <a href="doUser.jsp?o=delete&id={id }">delete</a>
            <input type="button" value="tests">
        </td>
    </tr>
</table>
</div>
    <div class="panelBar" style="width: 760px;" size="5,10,30,50"
    pagetheme="FULL" id="dataPageDiv">
    </div>

</form>

```

13.EasyDataTable 分页标签国际化支持

EasyDataTable 自带了分页标签 ,需要自定义显示的文字和语言时 ,标签中的文字可通过 language 参数调整和修改.

13.1 默认分页标签文字和语言

默认分页配置定义在DataTable.MSG属性中 ,默认分页标签文字和语言 :

```

DataTable.MSG={

    first:"首页",

    previous:"上一页",

    next:"下一页",

    last:"末页",

    totalCount:"共{0}条",

    totalPage:"共{0}页",

```

```
        rowPerPage: "每页显示{0}条"
    }
}
```

每页显示 5 条, 共 54 条

共 11 页 首页 上一页 1 2 3 4 5 6 7 下一页 末页 1 >>

13.2 自定义分页标签文字和语言

可直接修改和重新定义 DataTable.MSG 默认分页属性作为默认分页文字和语言。或者在初始化参数中专门指定（{0}为显示相应数据的占位符,必须存在）：

```
var pageLanguage={
    "first":'first',
    "previous":'previous',
    "next":'next',
    "last":'last',
    "totalPage":'total {0} pages',
    "totalCount":'total {0} rows',
    "rowPerPage":'page for {0} rows',
};

DataTable.load("datatable",{
    "pagetheme":DataTable.SIMPLE_PAGE,
    "loading":true,
    "language":pageLanguage
});
```

page for 5 rows , total 54 rows

total 11 pages first previous next last 1 >>