Jquery datalistview

1.1. CSS 文件

<link rel="stylesheet" href="datalistview/jquery.datalistview.css">
<link rel="stylesheet" href="datalistview/bootstrap.datalistview.css">

1.2. JS 文件

<script src="jquery-3.3.1.min.js"></script>

<script type="text/javascript" src="datalistview/jquery.datalistview.js"></script>
<script type="text/javascript" src="datalistview/common.js"></script>

2. HTML 代码片段

```
<div id="pager">
    <div id="info" class="pagination-info" style="float: right"></div>
</div>

<div id="data_listview"></div>

<div id="pager1">
    <div id="info1" class="pagination-info" style="float: right"></div>
</div>
```

3. 初始化参数(options)

- **header**: #是否显示表头
  value: true/false；type : bool
- **canselect**: #是否支持选中数据
  value：true/false；type：bool，
  如果不支持选中数据，需要定义 css 样式，设置背景色

```
.dlv-row-over, .dlv-row-selected, .dlv-row {
    background: #FFFFFF;
}
```

- **multi**: #是否可多选
  value：true/false；type：bool

- **pagination**　　#页码选项
  - pageDiv　#列表头部页码 div 容器
    value：div 选择器；type：string
  - pageInfo　#页码相关信息说明(样式：*显示第 x 到 y 条，共 z 条记录*)
    value：div 选择器；type: string
  - pageDiv1　#列表头部页码 div 容器
    value：div 选择器；type：string
  - pageInfo1 #页码相关信息说明(样式：*显示第 x 到 y 条，共 z 条记录*)
    value：div 选择器；type: string
    注：*pageDiv/pageDiv1 并不能决定哪个是头，哪个是尾，头尾由其在 html 代码中出现的顺序决定，pageDiv/pageInfo 也不必须是嵌套关系*
  - pageList　#单页可显式数据条数列表
    value：数据条数列表；type：list

  - pageSize　#默认每页展示的数据条数
    value：数据条数；type：int
    注：*如果填写的是 pageList 数组中的数字，在下拉菜单出现展示的数字，如果填写的不是 pageList 数组中的数字，下拉菜单中默认不会显示数字，但是每页显示的数据数量是和 pageSzie 相同的*

  - buttons　#自定义按钮和事件
    value：按钮定义；type：array
    例如：　　buttons = [{
    text : '',
    title : "删除",
    handler : function() {
    var selected = $("#all_news_listview").datalistview("selected");
    var ids = [];
    for (var i = 0; i < selected.length; i++) {
    var t = selected[i];
    ids.push(t._id + ":" + t.userid);
    }
    var max_cart_num = 20;
    if (ids.length > 0) {
    if (ids.length > max_cart_num) {
    alert("你选择的数据有点多");
    } else {
    delete_it(ids);
    }
    } else {
    alert("请先选择数据！");
    }
    }
    }];

- **sortSet** #数据排序选项
  value: 排序字段/顺序列表；type：列表
  如：[{
  id: “字段名”，
  sort: “asc” / “desc”
  }, …]
- **multiSort** #是否支持多字段联合排序
  value：true/false；type：bool
  注：默认为 true，如果为 false 则只能依据单个字段排序
- **columns**： #列表中的各个字段
  value：列的集合；type：list
  每个列都由以下四个选项定义：
  - id：字段标识，必须与后台返回的数据对应
  - title：显示在列表上的名字
  - align : "center" 对齐方式，left, right, center
  - width : 120 宽度 pixel，不指定则浏览器自己计算
  - sortAble：是否支持排序 非必须，默认为 false，当为 true 时，点击 title 会把 id sort 放入 sortSet 中进行排序
  - formatter：值为函数，控制该字段值的显示样式，非必须，默认为 title，函数定义如下：
    function(index, source)
    · index：当前为第几条数据
    · source：包含所有数据，通过 source.data[index]获取当前行的数据
    · 可以返回 html 或 jquery 对象
- **methods:** #某些事件的回调函数
  - onSelected: function(selected)：选中某条数据记录时会触发的函数
  - onAddRow: function(table, tr, data)：渲染数据记录行时会触发的函数
    例如：

```
onAddRow : function(table, tr, data) {
    if (data["del"] !== 0) {
        tr.addClass("deleted");
    }
}
```

- **source**： #数据源，可以直接返回 json 数据或 ajax 回调，函数定义如下：

```
function(data){
    $.ajax({
        url: 获取数据的 url,
        data:{
            "tableInfo": JSON.stringify(data),  #列表显示参数，如页码、排序方式等,
            key: value      #其他查询参数
        },
        dataType: “JSON”,
```

```
                success:function(data):{
                    $("#data_listview").datalistview("setSource", [data]);
                    $("#data_listview ").datalistview("loading", [true]);
                },
                error:function (xhr,code,what) {
                    $("#data_listview").datalistview("loading", [false]);
                }
    })
}
```

4. 实例方法（methods）

- **setSource** ： 设置 json 数据，ajax 返回后调用，json 格式为：{"keyid": "唯一标识名", "pagesize": pageSize, "offset": offset, "total": n, "data": [{ ... }, {...}] }，返回数据里面必须包含 keyid 对应的 "唯一标识名" 字段，例如 keyid = “id”, data 里面必须包含 id

  例：$("#data_listview").datalistview("setSource", [data]);

- **Loading**： 设置加载中或隐藏

  例：$("#data_listview ").datalistview("loading", [true]); 显示

  例：$("#data_listview").datalistview("loading", [false]); 隐藏

- **reload**: 手动重新刷新加载，调用 ajax 获取数据重新显示

  例：$("# data_listview").datalistview("reload")

- **selected :** 获取已选择数据，支持分页中多选

- **deselect :** 取消多选择数据

- **select** ： 选择多个，参数可以为 true，则全部选择， 或为 keyid 的值列表，数组 [ ]

5. 数据加载

首次加载：$("#data_listview").datalistview(query_options)
重新加载：$("# data_listview").datalistview("reload")

6. 后台代码(python)

```
def query(request):
    params = request.GET.dict()

    if not "tableInfo" in params:
        return HttpResponse(status=400)

    tableInfo = json.loads(params["tableInfo"])
```

```
    offset = int(tableInfo.get("offset", 0))
    pageSize = int(tableInfo.get("pageSize", 1))
    dao = MyData.objects.filter(...)  #根据后台的查询参数进行过滤查询

    sortSet = tableInfo.get("sortSet", [])

    sort_fields = []
    for asort in sortSet:
        sortid = asort["id"]
        sortdir = asort.get("sort", "desc")

        sort_fields.append("%s %s" % (sortid, "desc" if sortdir == "desc" else "asc"))

    dao = dao.order_by(", ".join(sort_fields))

    items = [obj.to_dict() for obj in dao[offset:offset + pageSize]]

    # keyid 为 obj.to_dict() 数据的主键，例如 id
    results = {"keyid": "id", "pagesize": pageSize, "offset": offset, "total": dao.count(), "data":
list(items)}

    return JsonResponse(results)
```