



链滴

Struts2 接收 DataTables 服务端处理参数 以及简单的分页模型

作者: [xuwangcheng14](#)

原文链接: <https://ld246.com/article/1484996069013>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```

<p>
    后端使用struts2的时候,Datatables使用服务端处理的话, 怎么处理自动发送的参数呢?
</p>
<p>
    结合实际开发,弄了一个比较简单易用的处理方法,仅适用于不是特别复杂的场景,另外附上分页模型。
</p>
<p>
    Datatables版本为1.10.0
</p>
<p>
    <br />
</p>
<p>
    <strong>struts2获取DT发送参数的工具类,可在action中调用</strong>
</p>
<p>
    <br />
</p>
<p>
    <br />
</p>
<pre class="prettyprint lang-java">/*
 * 获取DT发送的参数
 *
 */
public static Map<String,Object> getDTParameters(){
    Map<String,Object> returnMap = new HashMap<String,Object>();
    //排序的那一列位置
    String orderColumnNum = ServletActionContext.getRequest().getParameter("order[0][col
mn]");
    //排序方式 asc或者desc
    String orderType = ServletActionContext.getRequest().getParameter("order[0][dir]");
    //全局搜索条件
    String searchValue = ServletActionContext.getRequest().getParameter("search[value]");
    //需要排序的那一列属性名称
    String orderDataName = ServletActionContext.getRequest().getParameter("columns["+ord
rColumnNum+"][data]");
    //获取当前所有的展示字段
    Map<String, String[]> params = ServletActionContext.getRequest().getParameterMap(
;
    List<String> dataParams = new ArrayList<String>();
    for(Map.Entry<String, String[]> entry:params.entrySet()){
        if(entry.getKey().indexOf("[data]")!=-1){
            String a = (params.get(entry.getKey())[0];
            if(!a.equals("")){
                dataParams.add(a);
            }
        }
    }
    returnMap.put("orderDataName", orderDataName);
    returnMap.put("orderType", orderType);
    returnMap.put("searchValue", searchValue);
    returnMap.put("dataParams", dataParams);
    return returnMap;
}

```

```
}</pre>
<p>
  <br />
</p>
```

Action中对应的list方法

```
<p>
  <br />
</p>
<p>
  <b>备注:此为BaseAction中的方法,如果你没有ActionBase类把T改成对应实体类型就行了</b>
</p>
<p>
  <br />
</p>
<pre class="prettyprint lang-java">/**
 * 通用 list方法
 * 分页展示对应实体的集合
 * @return
 */
public String list(){
Map<String,Object> dt = StrutsUtils.getDTParameters();
PageModel<T> pu = baseService.findByPager(start, length,(String)dt.get("orderDataName"),(String)dt.get("orderType"),(String)dt.get("searchValue"),(List<String>)dt.get("dataParams"));
jsonMap.put("draw", draw);
jsonMap.put("data", pu.getDatas());
jsonMap.put("recordsTotal", pu.getRecordCount());
if(!((String)dt.get("searchValue")).equals("")){
jsonMap.put("recordsFiltered", pu.getDatas().size());
}else{
jsonMap.put("recordsFiltered", pu.getRecordCount());
}
jsonMap.put("returnCode", 0);
return SUCCESS;
}
</pre>
```

Dao方法中实现

```
<p>
  <strong>备注:同样为BaseDao,ORM框架为Hibernate</strong>
</p>
<p>
  <br />
</p>
```

```
<pre class="prettyprint lang-java">public PageModel<T> findByPager(int pageNo, int pageSize,String orderDataName,String orderType,String searchValue,List<String> dataParams) {
// TODO Auto-generated method stub
PageModel<T> pm=new PageModel<T>(orderDataName, orderType,searchValue,ataParams,dataNo,pageSize);
```

```
String hql="from "+clazz.getSimpleName();
```

```
//增加搜索条件
```

```
if(searchValue!=""){  
    hql+=" where ";  
    int i = 1;  
    for(String s:dataParams){  
        hql+=s+" like '%" + searchValue + "%'";  
        i++;  
        if(i<=dataParams.size()){  
            hql+=" or ";  
        }  
    }  
}
```

```
}
```

```
//增加排序
```

```
if(orderDataName!=""){  
    hql += " order by " + orderDataName + " " + orderType;  
}
```

```
pm.setDatas(getSession().createQuery(hql).setFirstResult(dataNo).setMaxResults(pageSize).setCacheable(true).list());  
pm.setRecordCount(totalCount());  
return pm;  
}  
</pre>
```

```
<br />
```

```
<p>  
    <strong>分页模型PageModel<T></strong>  
</p>
```

```
</p>
```

```
<p>  
    <br />
```

```
</p>
```

```
<pre class="prettyprint lang-java">/**
```

```
*
```

```
*/
```

```
package com.dcits.business.base.bean;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
/**
```

```
● @author xuwangcheng
```

```
● 分页模型
```

```
*/
```

```

public class PageModel<T>
{
//当前需要排序的列名称
private String orderDataName;
//排序方式,默认asc
private String orderType = "asc";
//全局搜索条件
private String searchValue = "";
//当前展示的字段名
private List<String> dataParams = new ArrayList<String>();
//当前起始数据的位置
private int dataNo=0;
//每页显示的记录数
private int pageSize=10;
//总记录数
private int recordCount;
//总页数
private int pageCount;
//存放分页数据的集合
private List<T> datas;
public PageModel(){
}
public PageModel(String orderDataName, String orderType,
String searchValue, List<String>
dataParams, int dataNo,
int pageSize) {
super();
this.orderDataName = orderDataName;
this.orderType = orderType;
this.searchValue = searchValue;
this.dataParams = dataParams;
this.dataNo = dataNo;
this.pageSize = pageSize;
}
public List<String> getDataParams() {
return dataParams;
}
}

```

```
public void setDataParams(List<String
gt; dataParams) {
this.dataParams = dataParams;
}
public String getOrderDataName() {
return orderDataName;
}
public void setOrderDataName(String orderDataName) {
this.orderDataName = orderDataName;
}
public String getOrderType() {
return orderType;
}
public void setOrderType(String orderType) {
this.orderType = orderType;
}
public String getSearchValue() {
return searchValue;
}
public void setSearchValue(String searchValue) {
this.searchValue = searchValue;
}
public int getDataNo() {
return dataNo;
}
public void setDataNo(int dataNo) {
this.dataNo = dataNo;
}
public int getPageSize() {
return pageSize;
}
public void setPageSize(int pageSize) {
this.pageSize = pageSize;
}
public int getRecordCount() {
return recordCount;
}
```

```
public void setRecordCount(int recordCount) {
    this.recordCount = recordCount;
}
public int getPageCount() {
    if(this.getRecordCount()&lt;=0){
        return 0;
    }else{
        pageCount=(recordCount+pageSize-1)/pageSize;
    }
    return pageCount;
}
public void setPageCount(int pageCount) {
    this.pageCount = pageCount;
}
public List&lt;T&gt; getDatas() {
    return datas;
}
public void setDatas(List&lt;T&gt;
datas) {
    this.datas = datas;
}
}
```

</pre>

<p>

</p>

<p>

本人为 Java新手,如有错误,欢迎指出并一起讨论!

</p>