

# table

定义示例

tabletablecoltablebutton

只列出了table的基本定义，属性见下面表格

```
<tag>
  <name>table</name>
  <tag-class>com.jxtech.tag.table.TableTag</tag-class>
  <body-content>JSP</body-content>
  <attribute>
    <name>id</name>
    <required>true</required>
    <rtexprvalue>>false</rtexprvalue>
  </attribute>
  <dynamic-attributes>true</dynamic-attributes>
</tag>
```

name	required	rtexprvalue	description
id	true	false	表格的唯一标识，整个输出的HTML文件中唯一，建议直接取jboname+_table，如果重复，再后面再添加1、2、3序列
label	false	false	表格标题说明性文字
selectmode	false	false	表格第一列显示选择框，取值范围：multiple、single、none
jboname	false	false	if relationship ==null 代表要查询的表名 else 代表父表的表名
relationship	false	false	联系名，联系需要在MaxRelationship表中配置，必须与jboname一起使用
select	false	true	需要查询的字段，默认为：*
groupby	false	true	分组的字段
orderby	false	false	排序字段
jboClassName	false	true	如果包含此属性，则优先于jboname属性，进行数据查询，主要用于虚拟表处理。
inputmode	false	false	表格的编辑模式，取值为： readonly：只读（默认） query：查询 edit：可编辑
datasrc	false	false	数据源
apprestrictions	false	false	查询的限制条件，例： status='CLOSE' AND Year='2013'，支持EL表达式

pagesize	false	false	默认的每页显示数据量，默认为20
trwhereattr	false	false	对应数据库字段
trwhere	false	false	运算符，可用值为： ==：等于 >：大于 <：小于
trwherevalue	false	false	数据库字段对应值， 与trwhereattr、trwhere联合使用
trfontcolor	false	false	字体颜色，可用值： 颜色名称的颜色（比如 red） 十六进制值的颜色（比如 #ff0000） rgb 代码的颜色（比如 rgb(255,0,0)） inherit 从父元素继承颜色。
trbgcolor	false	false	背景颜色，可用值： 颜色名称的颜色（比如 red） 十六进制值的颜色（比如 #ff0000） rgb 代码的颜色（比如 rgb(255,0,0)） inherit 从父元素继承颜色。
bottomtipvalue	false	false	底部小提示内容， 字体颜色默认为红色，可用bottomtipcolor属性设置字体颜色 该属性支持html标签，如有需要 可在此嵌入html标签
bottomtipcolor	false	false	底部小提示字体颜色，可用值： 颜色名称的颜色（比如 red） 十六进制值的颜色（比如 #ff0000） rgb 代码的颜色（比如 rgb(255,0,0)） inherit 从父元素继承颜色。
initWhereCause	false	false	初始化条件，可以用逗号分隔多个条件，每个条件用AND连接。此条件与apprestrictions的区别是，initWhereCause可以被覆盖，apprestrictions不能被覆盖
initWhereParam	false	false	初始化条件中的参数值，必须配合initWhereCause一起使用，可以用逗号分隔多个条件。

initOverlay	false	true	初始化条件中的参数值是否可以覆盖。取值为true/false
url	false	true	重新加载数据时，从哪里加载，默认：app.action
type	false	true	加载表格的类型（JSP页面），默认为：list-table
title	false	true	表格名称，如果为空，则不显示表头中及表头的上一条、下一条等信息。如果只想显示上一条、下一条则可以设置为空格字符。
footVisible	false	true	是否显示页脚，默认true
tabletype	false	true	list: （默认）列表 imglist: 图片列表，当为此值时，第一个tablecol必须为图片地址
imgcols	false	true	显示图片列表时，每行显示多少张图片
imgwidth	false	true	显示图片列表时，每个图片的宽度，默认为auto
imgheight	false	true	显示图片列表时，每个图片的高度，默认为auto
imgtype	false	true	当type=imglist有效，为空（默认）显示原始图片。 mini: 显示缩略图 watermark: 显示水印图
filterable	false	true	"true": 显示快速过滤。默认不需要。 当列表存在tablecol的visibleHead属性为true时，该属性失效
height √ desktop mobile忽略此属性	false	true	该属性用于设置dataTable 优先级最高的固定表格高度属性，当ignoreDataTable为true时该属性失效
fixedWidth √ desktop mobile忽略此属性	false	true	该属性用于设置dataTable 数字即可，当表格太宽，给一个比较大一点的固定宽度吧，推荐只在列表页面使用，详细页面使用不好看啊。 不再建议使用，table现在会自动应用table-layout:fixed，详情参考fragment的displayMode属性
ignoreLayoutFixed √ desktop mobile忽略此属性	false	true	使用该属性会取消对表格第一行tr为标题栏的表格进行宽度设定。默认为false

ignoreDataTable ✓desktop mobile忽略此属性	false	true	所有list-table类表格默认都会使用dataTable进行界面调整(确保头部和底部都在可视范围内),如果你确认你的表格不需要进行界面调整,将该属性设为true。默认为false。
sql	false	true	直接写SQL脚本,注意,当Jboname以SQL_开头时,此处的SQL脚本才有效,即jboname需要设置一个临时的名称。
rowSelectable *new	false	true	是否允许选中行。当选中行时候会选中改行所包含的checkbox
executeQueryAfterLoad	false	true	是否执行Jbo中的afterLoad方法,默认为执行,false为不执行。



备注

1. trwhereattr、trwhere、trwherevalue必须联合使用,组成一个表达式返回一个布尔值。如果为真trfontcolor和trbgcolor才生效。

只列出tablecol的基本定义,属性见下面表格

```
<tag>
  <name>tablecol</name>
  <tag-class>com.jxtech.tag.table.TablecolTag</tag-class>
  <body-content>JSP</body-content>
  <attribute>
    <name>id</name>
    <required>>false</required>
    <rtexprvalue>>false</rtexprvalue>
  </attribute>
  <dynamic-attributes>>true</dynamic-attributes>
</tag>
```

name	required	rtexprvalue	description
id	false	false	
dataattribute	false	false	字段名, 联系名.字段名 均可,参见示例1 可以多个字段合并,使用PL/SQL语法(参见示例4) ROWNUM, 直接显示序号 自定义列
datatype	false	true	自定义类型,覆盖数据库中配置的类型。类型参见: <a href="#">JXADF数据类型定义</a>
secondAttributes	false	true	可以填写多个属性,自动在TD中生成"属性=值"

dataDisplay	false	true	可以根据dataattribute中的值，显示本翻译之后的值，本处需要使用JSON格式，参见例7：  dataDisplay="{ '1': 'green.gif', '2': 'dynamic-red.gif' }"
dataname	false	true	仅针对输出JSON数据有效，JSON的字段名。
label	false	false	该列的标题（th）名称
mxevent	false	false	取值为：  SELECTRECORD:使用本平台打开当前记录的详细信息  WORKFLOW: 打开工作流程历史信息  ATTACHMENT:附件链接  LINK: 链接  DELOW:删除当前记录  LOOKUP: 查询  其它值: 将自动生成一个按钮，事件为: mxevent(this,event,uid)，例: mxevent=start，则生成start(this,event,uid)事件，需要自己实现start方法。
mxevent_render	false	false	当mxevent= <b>其他值</b> 时生效  INPUT: 默认值，用button的形式来渲染，默认值  SWITCH:用开关形式来渲染  LINK:用A标签来渲染
mxevent_desc	false	false	当mxevent=WORKFLOW时，这里值为: lcam/project  当mxevent=ATTACHMENT时，这里为表名（TOP_ATTACHMENT）. 字段名（JOB_TYPE_CODE）的值，注意大小写敏感，建议为表名
mxevent_icon	false	false	图标文件名，默认为document.gif
mxevent_disabled	false	false	mxevent设定值之后，生成的按钮是否可以点击。支持EL表达式。 示例参见: <a href="#">table ex05</a>
linkedcontrolid	false	false	
lookup	false	false	参见lookup功能
lookupWrite	false	false	配合lookup功能 是否能写input中内容 true可写 false不可写
filterable	false	false	可过滤（true）、不能过滤（false）
filtername	false	true	过滤字段名，为空则取dataattribute

sortable	false	false	<p>是否有点击排序功能（默认true），取值为：</p> <p>true: 添加排序函数</p> <p>false: 不添加排序函数</p>
sortname	false	true	排序字段名，如果为空，则默认为dataattribute
type	false	true	<p>默认为空，当tablecol包含pushbutton时,设置type="btn_group"将在mobile中显示按钮组的效果。</p> <p>当MXEVENT为其它值时，要使用链接样式，这里可以设定为 LINK</p>
startHtml	false	false	插入值之前的HTML代码
html	false	false	单元格（td）的内容,这里可以写html标签，插入当前值之后的HTML代码
urlType	false	false	<p>URL 的类型，目前支持 comtop,report,自定义URL地址, mxevent=LINK时有效</p> <p>comtop: 需要在WEB-INF\base.properties中配置jx.comtop.url</p> <p>report: 需要在WEB-INF\base.properties中配置jx.report.url</p> <p>me: 会自动添加当前应用的前缀，例: http://127.0.0.1:8080/jxweb</p> <p>自定义URL地址: 以http://开头的地址</p>
urlParamName	false	false	<p>参数名，与urlType一起使用有效，跟在URL后面的参数</p> <p>支持ELUtil处理的el表达式（针对jbo对象的），比如: <code>urlParamName= "\${APP}/index.action"</code></p>
urlParamValue	false	false	<p>参数值，与urlParamName一起使用，跟在URL后面的参数。</p> <ol style="list-style-type: none"> <li>单独使用。不需要json格式，只需要传入表字段名。例如： <code>urlParamValue="ITEM_ID"</code>。</li> <li>传入的为类json格式的数据。 会自动转换成键值对拼凑在request url后面。 例如： <code>urlParamValue="userId:USER_ID,jobId:JOB_ID"</code></li> </ol> <p><b>注意：</b> <code>urlParamValue</code>的值不需要<code>{}</code>，不需要单引号。 参见示例3</p>
urlTarget	false	false	与link或urlType一起使用有效，超链接打开的目标地址。

format	false	false	<p>为数字时有效，目前支持以下格式</p> <table border="1"> <thead> <tr> <th>值</th> <th>示例值</th> <th>示例显示值</th> </tr> </thead> <tbody> <tr> <td>,###.00</td> <td>123456.789</td> <td>123,456.79</td> </tr> <tr> <td>,###</td> <td>123456.789</td> <td>123,456.789</td> </tr> <tr> <td>,###.0</td> <td>123456.789</td> <td>123,456.8</td> </tr> <tr> <td>\$</td> <td>123456.789</td> <td>\$123,456.79</td> </tr> <tr> <td>¥</td> <td>123456.789</td> <td>¥123,456.79</td> </tr> <tr> <td>rmb</td> <td>123456.789</td> <td>¥123,456.79</td> </tr> <tr> <td>%</td> <td>21.2</td> <td>21.2%</td> </tr> <tr> <td>0.00%</td> <td>21.2</td> <td>21.20%</td> </tr> </tbody> </table>	值	示例值	示例显示值	,###.00	123456.789	123,456.79	,###	123456.789	123,456.789	,###.0	123456.789	123,456.8	\$	123456.789	\$123,456.79	¥	123456.789	¥123,456.79	rmb	123456.789	¥123,456.79	%	21.2	21.2%	0.00%	21.2	21.20%
值	示例值	示例显示值																												
,###.00	123456.789	123,456.79																												
,###	123456.789	123,456.789																												
,###.0	123456.789	123,456.8																												
\$	123456.789	\$123,456.79																												
¥	123456.789	¥123,456.79																												
rmb	123456.789	¥123,456.79																												
%	21.2	21.2%																												
0.00%	21.2	21.20%																												
inputsize	false	true	如果为输入框有效，输入框的大小																											
maxlength	false	true	最多显示多少个字符，超过用省略号代替。																											
width ✓desktop mobile忽略此属性	false	true	列宽																											
height ✓desktop mobile忽略此属性	false	true	行高																											
required	false	true	是否必填，false/true																											
note	false	true	备注，展示在列名下																											
readonly	false	true	行编辑时该列只读																											
visible	false	true	是否展示此列，默认展示 true: 展示 false: 不展示																											
visibleHead ✓desktop mobile忽略此属性	false	true	是否显示列头，如果为false，则将此列与前一列合并显示在一列中。																											
render	false	true	渲染， TEXT: 默认，直接按文本显示 HTML: 按HTML标记显示 LABELS: 以空格分隔的多个标签，可点击链接查询																											

cssClass	false	true	额外的校验样式，详细参见validations
----------	-------	------	-------------------------

## 嵌套select控件

参考: select

基本定义
<pre> &lt;tag&gt;   &lt;name&gt;tablebutton&lt;/name&gt;   &lt;tag-class&gt;com.jxtech.tag.table.TableButtonTag&lt;/tag-class&gt;   &lt;body-content&gt;JSP&lt;/body-content&gt;   &lt;attribute&gt;     &lt;name&gt;id&lt;/name&gt;     &lt;required&gt;&gt;false&lt;/required&gt;     &lt;rtexprvalue&gt;&gt;true&lt;/rtexprvalue&gt;   &lt;/attribute&gt;   &lt;dynamic-attributes&gt;&gt;true&lt;/dynamic-attributes&gt; &lt;/tag&gt; </pre>

name	required	rtexprvalue	说明
id	false	true	命名规范: 表格ID+中划线+事件名  如果将ID注册到MaxMenu表中, 就可进行权限检查, 未注册, 则不进行权限检查。
label	false	true	标签
mxevent	false	true	事件, JS事件名, 例: add, 则生成add(this,event)  href: 直接链接地址, 链接地址为params
mxtype	false	true	
disabled	false	true	按钮是否活动, 一般不用设置, 由JBO决定
params	false	true	参数
url	false	true	URL参数
icon	false	true	如果不为空, 则显示图标



## 将ID注册到Maxmenu表中示例

-- 角色权限管理功能中，将角色是否可以新增用户范围按钮进行权限控制示例。子表的列表新增按钮

--SQL脚本

```
insert into
maxmenu(maxmenuid,app,menu,description,menutype,position,parent,url,accesskey,image,tabdisplay,visible,
tags,EXTENDS,MEMO,APPTYPE)
values
(maxmenuidseq.nextval,'PUBROLE','userscopelist-addrow','新增用户范围','OPTION',1100,null,null','','','NO
NE',1','','',NULL,'MAIN');
```

--FTL脚本

```
<@jxui.tablebutton label='${lang["table.PubRoleUserScope.button.add"]}'
id="userscopelist-addrow" mxevent="addrow" mxtype="icon"/>
```

Base参数字段滤镜SQLButton附件JSON校验



说明

显示检查计划列表

## JSP页面中的用法

```
<jxui:table jboname="PS_PGC_SAFE_CKPLAN" id="ps_pgc_safe_ckplan_table"
trwhereattr="WFT_BACKFLAG" trwhere="==" trwherevalue="T"
trfontcolor="red" trbgcolor="rgb(241,237,237)" selectmode="multiple"
bottomtipvalue="注：红色表示回退的单据。" bottomtipcolor="red"
apprestrictions="valid_status is not null">
<jxui:tablecol dataattribute="CKPLAN_NUM" type="link" mxevent="selectrecord"/>
<jxui:tablecol dataattribute="CKPLAN_NAME" />
<jxui:tablecol dataattribute="CKUSER_NAME" />
<jxui:tablecol dataattribute="TIME_LIMIT" />
<jxui:tablecol dataattribute="START_DATE" />
<jxui:tablecol dataattribute="END_DATE" />
<jxui:tablecol dataattribute="PLAN_HERALD" />
</jxui:table>
```

## 插件中FTL页面的用法

```
<@jxui.table jboname="PS_PGC_SAFE_CKPLAN" id="ps_pgc_safe_ckplan_table"
trwhereattr="WFT_BACKFLAG" trwhere="==" trwherevalue="T"
trfontcolor="red" trbgcolor="rgb(241,237,237)" selectmode="multiple"
bottomtipvalue="注：红色表示回退的单据。" bottomtipcolor="red"
apprestrictions="valid_status is not null">
<@jxui.tablecol dataattribute="CKPLAN_NUM" type="link" mxevent="selectrecord"/>
<@jxui.tablecol dataattribute="CKPLAN_NAME" />
<@jxui.tablecol dataattribute="CKUSER_NAME" />
<@jxui.tablecol dataattribute="TIME_LIMIT" />
<@jxui.tablecol dataattribute="START_DATE" />
<@jxui.tablecol dataattribute="END_DATE" />
<@jxui.tablecol dataattribute="PLAN_HERALD" />
</@jxui.table>
```




说明

以上两种写法，分别对应不同的文件类型以及开发方式，其它例子只采用其中一种写法，如果用途不一样，请自行转换。

## 更多

```
<jxui:tablecol dataattribute="CKPLAN_NUM" type="link" mxevent="selectrecord"/>
<jxui:tablecol dataattribute="CKPLAN_NAME" />
<jxui:tablecol dataattribute="PS_PGC_SAFE_CKPLANCKUSER_ID.NAME" label="检查负责人"/>
<jxui:tablecol dataattribute="TIME_LIMIT" />
<jxui:tablecol dataattribute="START_DATE" />
<jxui:tablecol dataattribute="END_DATE" />
<jxui:tablecol dataattribute="PLAN_HERALD" />
<jxui:tablecol label="删除" sortable="false" mxevent="delrow"/>
<jxui:tablecol label="自定义列测试" html="<a href='del'>自定义链接</a>" sortable="false" />
<jxui:tablecol dataattribute="wft_transactor" align="center" maxlength="7" type="link"
    mxevent="workflow" mxevent_desc="lcam/project" />
```

## 示例1

 当登录用户的project\_id为空时，整个条件解析结果为：PLAN\_TYPE=1，否则解析为 PLAN\_TYPE=1 and object\_id='000001'

查询限制条件中添加EL表达式，仅限于基建项目使用

```
<jxui:table jboname="PS_SCHEDULE_PLAN" id="listtable" selectmode="none"
    apprestrictions="PLAN_TYPE=1 ${jxfn:userProjectCause(' and project_id='')}">
  <jxui:tablecol dataattribute="PLAN_NAME" type="link" mxevent="selectrecord"/>
  <jxui:tablecol dataattribute="PS_SCHEDULE_PLANOBJECT_ID.Project_name" label="项目名称"
/>
  <jxui:tablecol dataattribute="PS_SCHEDULE_PLAN_INSURE_STATUS.item_name" label="有效?"/>
</jxui:table>
```

## 示例2

**i** ownerid,pnum为URL地址后面直接添加的参数名称，即获取URL中的参数方法如下

### 在插件平台中示例

```
<#assign wherec="">
<#assign wherep="">
<#if (parameters.ownerid??) && (parameters.ownerid[0]??)>
  <#assign wherec="ownerid= ? ">
  <#assign wherep=parameters.ownerid[0]>
</#if>
<#if (parameters.pnum??) && (parameters.pnum[0]??)>
  <#if (wherep!="")>
    <#assign wherep=wherep+", " >
    <#assign wherec=wherec+", " >
  </#if>
  <#assign wherec=wherec+"projectnum= ? ">
  <#assign wherep=wherep+parameters.pnum[0]>
</#if>
<@jxui.table jbone="IDR" id="listtable"
  initWhereCause="{wherec!}" initWhereParam="{wherep!}"
  selectmode="multiple" orderby="startDate desc" initOverlay="TRUE">
  <@jxui.tablecol dataattribute="IDROWNERID.Displayname"
label='${lang["table.idr.ownerid"]}'
  mxevent="link" urlType="me" urlParamName="idr/index.action?ownerid="
  urlParamValue="ownerid" urlTarget="o"/>
  <@jxui.tablecol dataattribute="title" label='${lang["table.idr.title"]}' mxevent="link"
urlType="me"
  urlParamName="idr/index_main.action?view=readonly&v=v&uid="
urlParamValue="IDRID"/>
  <@jxui.tablecol dataattribute="startdate" label='${lang["table.idr.startdate"]}'/>
  <@jxui.tablecol dataattribute="taskhour" label='${lang["table.idr.taskhour"]}'/>
  <@jxui.tablecol dataattribute="jira" label='${lang["table.idr.jira"]}' mxevent="link"
  urlType="http://jira.jxtech.net/browse" urlParamName="/" urlParamValue="JIRA"
urlTarget="j"/>
  <@jxui.tablecol dataattribute="IDRPROJECTNUM.Projectname"
label='${lang["table.idr.projectname"]}'
  mxevent="link" urlType="me" urlParamName="jxproject/index_main.action?uid="
  urlParamValue="IDRPROJECTNUM.projectid" urlTarget="p"/>
</@jxui.table>
```

### 示例3

**i** 连接报表, Table中使用EL表达式, 详细参见: report.jsp

report.jsp

```
<jxui:table jboname="ps_project" id="reporttable" selectmode="none"
           apprestrictions="{sessionScope.jxuserinfo.projectCause}">
  <jxui:tablecol dataattribute="PROJECT_NAME" mxevent="link" urlType="report"
                urlParamName="whskz_report.jsp?projectId" urlParamValue="project_id"
                urlTarget="_report"/>
</jxui:table>
```

urlParamValue的键值对用法 report.jsp

```
<jxui:tablecol dataattribute="PS_PGC_COST_PAYMENTSERVICE_CONTRACT_ID.service_contract_code"
              label="付款申请编号" mxevent="link" urlType="comtop"

              urlParamName="/lcam/project/pgcproject/costmanage/costPaymentListMultiOperateAction.do?action
              Type=InitInsert&"

              urlParamValue="paymentId:PAYMENT_ID,serviceContractId:SERVICE_CONTRACT_ID" sortable="false"/>
```

### 示例1

**i** dataattribute属性由多个字段拼接

list-table.jsp

```
<jxui:tablecol dataattribute="BID_YEAR||'年'||BID_MONTH||'月'" label="月度"/>
```

**i** 根据dataattribute中计算出来的值, 显示自定义的信息, 如图片、文字、HTML等, 注意图片如果在skin/xxx/images目录下, 可以直接写图片名称, 否则需要写img的HTML标签。

/jxweb/WEB-INF/content/app/firplantra/list-table.jsp

详细参见: 一级进度计划跟踪

```
<jxui:tablecol label="受控状态" align="center" dataattribute="CONTROLLED_STATUS"
              dataDisplay="{ '1': 'green.gif', '2': 'dynamic-red.gif' }"/>
```

## 格式化

**i** 数字格式化, 货币

list-table.jsp

```
<jxui:tablecol dataattribute="PROJECT_INVEST" label="概算总投资(万元)" format="¥"/>
```

## 虚拟字段

**i** 在Jbo中, 只要实现getXxxxx()即可, 例

jbo

```
public String getStatus() throws JxException {  
    //todo: 做你自己的事  
    return "I Love You";  
}
```

ftl

```
<jxui:tablecol dataattribute="status" />
```

## 多字段自由组合

 ons-table.ftl

```
ons-table.ftl

<@jxui.table jboname="JXOSGI" id="listtable" visibleHead="true" title=" " visibleHead="false"
cssClass="list">
  <@jxui.tablecol dataattribute="name" sortable="false"
startHtml="<table><tr><td class='listTitle'><button type='mybutton'
class='mybutton gray'>"
html="</button></td>"/>
  <@jxui.tablecol dataattribute="Version" visibleHead="false"
startHtml='<td class="listVer">${lang["table.jxosgi.version"]}:'
html='</td>' />
  <@jxui.tablecol dataattribute="comments" label='${lang["table.jxosgi.comments"]}'
visibleHead="false"
startHtml='<td class="listVer">${lang["table.jxosgi.comments"]}:'
html='</td>' />
  <@jxui.tablecol dataattribute="hits" label='${lang["table.jxosgi.hits"]}'
visibleHead="false"
startHtml='<td class="listVer">${lang["table.jxosgi.hits"]}:'
html='</td></tr>' />
  <@jxui.tablecol dataattribute="description" label='${lang["table.jxosgi.description"]}'
visibleHead="false"
startHtml="<tr><td colspan='100'>"
html="</tr></table>"
render="html"
/>
</@jxui.table>
```

## secondAttributes

 index-table.ftl

```
index-table.ftl

<@jxui.tablecol dataattribute="STATUS" label='' visibleHead="false"
secondAttributes="jxosgiid,downloadurl,server='${osgiServer}'"
startHtml='<td class="listVer"><span onclick="remoteAction(this,event)"
class="remoteAction">'
html='</span></td></tr>' />
```

## mxevent



### 注意

mxevent\_disabled支持EL表达式，EL表达式的写法很有讲究。

由于EL表达式，需要在标签中去解析，而不是第一次遇到就解析，所以要对表达式进行转义。

### mxevent 表达式写法

```
<!-- 直接使用jbo.getRolepermissiongrantmanagestatus() 的情况,必须全部小写-->
<@jxui.tablecol dataattribute="RolePermissionGrantManage"
    label='${lang["table.pub_role_operation.RolePermissionGrantManage"]}'
    mxevent="actGrantManage" dataDisplay="{ '1': 'YES', '0': 'NO' }"
    mxevent_disabled="{r'${jbo.rolepermissiongrantmanagestatus}'}/>

<!-- 要取jbo.getObject(attributename)中的值,写法如下 -->
<@jxui.tablecol dataattribute="RolePermissionGrantManage"
    label='${lang["table.pub_role_operation.RolePermissionGrantManage"]}'
    mxevent="actGrantManage" dataDisplay="{ '1': 'YES', '0': 'NO' }"
    mxevent_disabled="{r'${jbo.data.gmstatus}'}/>
```



### 示例

使用filterable来快速过滤表格数据，

参考：[svn/清蓄施工现场检查与整改移动平台/开发库/源代码/JxPlatform/PlatformSrc/jxweb/WEB-INF/content/app/servicecontract/condservercontract/lookup\\_list-table.jsp](#)

### 示例

```
<jxui:table jboname="PS_SERVICE_CONTRACT" id="listtable" selectmode="none" filterable="true">
  <jxui:tablecol label="选择" dataattribute="SERVICE_CONTRACT_ID" mxevent="lookup"/>
  <jxui:tablecol dataattribute="SERVICE_CONTRACT_CODE" />
  <jxui:tablecol dataattribute="SERVICE_CONTRACT_NAME" />
  <jxui:tablecol dataattribute="VENDOR_COMPANY" label="承包商"/>
</jxui:table>
```



### 概述

直接使用SQL脚本

参考：[myidr-month-table.ftl](#)

### ftl

```
<#assign jxui=JspTaglibs["/WEB-INF/tlds/jxui.tld"] />
<@jxui.table jboname="SQL_IDR_YEAR" id="idryear" selectmode="none" orderby="startdate desc"
  sql="select a.projectnum, sum(taskhour)/8 taskhour, startdate,b.projectname
  from (select projectnum, taskhour, to_char(startdate, 'yyyy') startdate
  from idr where ownerid='${jxuserinfo.userid}') a, jxproject b
  where a.projectnum=b.projectnum and taskhour>0
  group by a.projectnum, startdate,b.projectname ">
  <jxui.tablecol dataattribute="startdate" label='${lang["table.idr.startdate"]}' />
  <jxui.tablecol dataattribute="taskhour" label='${lang["table.idr.days"]}' />
  <jxui.tablecol dataattribute="projectnum" label='${lang["table.idr.projectnum"]}' />
  <jxui.tablecol dataattribute="projectname" label='${lang["table.idr.projectname"]}' />
</@jxui.table>
```

## button


 在页面的子表中显示新建按钮。详细参见：[ps\\_schedule\\_task-list.jsp](#)

ps\_schedule\_task-list.jsp

```
<jxui:table id="ps_schedule_task" relationship="PS_SCHEDULE_TASK_PLAN_ID"
  jbname="PS_SCHEDULE_PLAN" orderby="to_number(task_code)" title="计划明细" inputmode="edit">
<jxui:tablebutton id="ps_schedule_task-addrow" label="新建" mxevent="addrow" mxtype="icon"/>

<jxui:tablecol dataattribute="TASK_CODE" label="序号" />
<jxui:tablecol dataattribute="TASK_NAME" label="任务名称" />
<jxui:tablecol dataattribute="PLAN_FINISH_DATE" />
<jxui:tablecol dataattribute="REMARK" />
<jxui:tablecol mxevent="delrow" label="删除"/>
</jxui:table>
```

## 附件

 打开的附件页面是否只读，可以通过 重载 `Jbo.isReadOnly()` 方法来实现。

```
<jxui:tablecol label="附件" mxevent="attachment" mxevent_desc="PS_PGC_WHS_CKPLAN_FILE"/>
```

## 定义

参数 jsonHead jsonFoot



**i** 对任意一个table的页面，在URL后添加参数render=json即可返回JSON数据格式。

对于JSON组件实现的内容，在本TABLE组件中都有实现，所以可以直接将现有的TABLE转为JSON格式，而不必再写一遍页面。

注意：“联系名.字段名”，如果没有配置dataname属性，将直接转换为“联系名\_字段名”。

参数	默认值	说明
render		=json时，输出JSON格式
pagesize	20	每页大小
pagenum	1	第几页
startStr		JSON的开始字符串，可以配置{、callback函数名等
endStr		JSON的结束字符串，可以配置}、)等。
jsonhead		示例: [{"total": "count"}, {"fixText": "\rows\":"}] 针对通用的table标签，如果table标签中已添加了jsonHead标签，则此处不需要再重复了。

tld

```
<tag>
  <name>jsonhead</name>
  <tag-class>com.jxtech.tag.json.JsonHeadTag</tag-class>
  <body-content>JSP</body-content>
  <attribute>
    <name>attributeLabel</name>
    <required>>false</required>
    <rtextprvalue>>true</rtextprvalue>
  </attribute>
  <dynamic-attributes>>true</dynamic-attributes>
</tag>
```

### 属性

name	required	rtextprvalue	描述
attributeLabel	false	true	属性标签
attribute	false	true	属性名称，即jboaset.getXXX
fixText	false	true	显示固定文本信息

## tld

```
<tag>
  <name>jsonfoot</name>
  <tag-class>com.jxtech.tag.json.JsonFootTag</tag-class>
  <body-content>JSP</body-content>
  <attribute>
    <name>attributeLabel</name>
    <required>>false</required>
    <rtexprvalue>>true</rtexprvalue>
  </attribute>
  <dynamic-attributes>>true</dynamic-attributes>
</tag>
```

## 属性

name	required	rtexprvalue	描述
attributeLabel	false	true	标签描述
attribute	false	true	属性，即jboaset.getXXX
fixText	false	true	固定文本

## 示例

直接使用添加头

---

**i** 无需任何修改，直接使用。

注意tablecol中的urlParamValue也做了相应的解析。

FTL页面不用做任何修改，只需要在访问的URL中添加参数即可。

```
<#assign jxui=JspTaglibs["/WEB-INF/tlds/jxui.tld"] />
<@jxui.table jboname="JXDataSync" id="listtable" selectmode="multiple">
  <@jxui.tablecol dataattribute="DataSyncNum"
label='${lang["table.jxdatasync.dataSyncNum"]}' mxevent="link"
          urlType="me" urlParamName="jxdatasync/index_main.action?uid="
urlParamValue="JxDataSyncid"/>
  <@jxui.tablecol dataattribute="Description"
label='${lang["table.jxdatasync.description"]}' />
  <@jxui.tablecol dataattribute="JXDATASYNCDBSOURCE1.description"
label='${lang["table.jxdatasync.dbSource1"]}' />
  <@jxui.tablecol dataattribute="JXDATASYNCDBSOURCE2.description"
label='${lang["table.jxdatasync.dbSource2"]}' />
  <@jxui.tablecol dataattribute="table1" label='${lang["table.jxdatasync.table1"]}' />
  <@jxui.tablecol dataattribute="table2" label='${lang["table.jxdatasync.table2"]}' />
  <@jxui.tablecol dataattribute="SyncDirection"
label='${lang["table.jxdatasync.syncDirection"]}'

dataDisplay='{0: '无需同步', 1: '数据源1->数据源2', 2: '数据源2->数据源1', 4: '数据源1<->数
据源2'}' />
  <@jxui.tablecol dataattribute="SyncTime" label='${lang["table.jxdatasync.syncTime"]}' />
</@jxui.table>
```

测试结果： /jxweb/jxdatasync/index\_list-table.action?render=json

```
[
  {
    "datasyncnum": "1025",
    "JxDataSyncid": "66",
    "description": "增量同步测试",
    "jxdatasyncdbsource1_description": "",
    "jxdatasyncdbsource2_description": "",
    "table1": "maxapps",
    "table2": "maxmodules",
    "syncdirection": "1",
    "synctime": ""
  },
  {
    "datasyncnum": "1001",
    "description": "同步测试",
    "JxDataSyncid": "1",
    "jxdatasyncdbsource1_description": "",
    "jxdatasyncdbsource2_description": "",
    "table1": "maxapps",
    "table2": "maxmodules",
    "syncdirection": "1",
    "synctime": "2014-12-25 16:13:48"
  }
]
```

**i** 添加 jsonHead、jsonFoot 的内容，将不会影响原来的表格渲染，只针对JSON输出格式有效，可参考JSON组件。

本例还添加了callback的测试，具体如下所示：

添加jsonhead标签，不影响原来的table渲染

```
<@jxui.table jboname="JXDataSync" id="listtable" selectmode="multiple">
  <@jxui.jsonhead attributeLabel="total" attribute="count"/>
  <@jxui.jsonhead fixText="rows:"/>

  <@jxui.tablecol dataattribute="DataSyncNum"
label='${lang["table.jxdatasync.dataSyncNum"]}'
          mxevent="link" urlType="me"
urlParamName="jxdatasync/index_main.action?uid=" urlParamValue="JxDataSyncid"/>
  <@jxui.tablecol dataattribute="Description"
label='${lang["table.jxdatasync.description"]}'/>
  <@jxui.tablecol dataattribute="JXDATASYNCDBSOURCE1.description"
label='${lang["table.jxdatasync.dbSource1"]}'/>
  <@jxui.tablecol dataattribute="JXDATASYNCDBSOURCE2.description"
label='${lang["table.jxdatasync.dbSource2"]}'/>
  <@jxui.tablecol dataattribute="table1" label='${lang["table.jxdatasync.table1"]}'/>
  <@jxui.tablecol dataattribute="table2" label='${lang["table.jxdatasync.table2"]}'/>
  <@jxui.tablecol dataattribute="SyncDirection"
label='${lang["table.jxdatasync.syncDirection"]}'

dataDisplay='{0: '无需同步', '1: '数据源1->数据源2', '2: '数据源2->数据源1', '4: '数据源1<->数
据源2}'/>
  <@jxui.tablecol dataattribute="SyncTime" label='${lang["table.jxdatasync.syncTime"]}'/>
</@jxui.table>
```

```
/index_list-table.action?render=json&pagesize=3&pagenum=1&startStr=mycallback({
&endStr=})
```

```
mycallback(
{
  "total": "2",
  "rows": [
    {
      "datasyncnum": "1025",
      "JxDataSyncid": "66",
      "description": "增量同步测试",
      "jxdatasyncdbsource1_description": "",
      "jxdatasyncdbsource2_description": "",
      "table1": "maxapps",
      "table2": "maxmodules",
      "syncdirection": "1",
      "synctime": ""
    },
    {
      "datasyncnum": "1001",
      "JxDataSyncid": "1",
      "description": "同步测试",
      "jxdatasyncdbsource1_description": "",
      "jxdatasyncdbsource2_description": "",
      "table1": "maxapps",
      "table2": "maxmodules",
      "syncdirection": "1",
      "synctime": "2014-12-25 16:13:48"
    }
  ]
}
)
```



### 摘要

演示字段的校验，更多参见：

1、validations

2、textbox

## 日期校验



### 示例说明

在请假表中，请假开始时间小于结束时间。

注意此处使用cssClass中的校验，按标准应为ID值，但这里由于是多行，无法设定ID值，请设定“dataattribute=”代替ID的前缀，需要注意大小写。

leave-line-table.ftl

```
<@jxui.tablecol dataattribute="BEGINDATE" label='${lang["table.leaveline.BEGINDATE"]}'
required = "true"/>
  <@jxui.tablecol dataattribute="ENDDATE" label='${lang["table.leaveline.ENDDATE"]}'
required = "true"
  cssClass="future[#dataattribute=BEGINDATE]"/>
```

### 请假明细

新增

假别	开始日期	结束日期	天数	删除
病假	2015-10-30	<span style="color: red;">* 日期不能小于 2015-11-2</span>	2	×
产假	2015-11-02	2015-10-01	10	×

共 2 条 每页 20 ( 20,50,80,100) 条 1 / 1页 转到 1 页 GO

## 问题处理

1. 列表界面查询条件会记忆，需要设定不记忆条件