快速使用

创建一个table实例最简单的方式是，在页面放置一个元素 *<table id="demo"></table>*，然后通过 *table.render()* 方法指定该容器



1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="utf-8">
5. <title>table模块快速使用</title>
6. <link rel="stylesheet" href="/layui/css/layui.css" media="all">
7. </head>
8. <body>
10. <table id="demo" lay-filter="test"></table>
12. <script src="/layui/layui.js"></script>
13. <script>
14. layui.use('table', function(){
15. var table = layui.table;
17. //第一个实例
18. table.render({
19. elem: '#demo'
20. ,height: 312
21. ,url: '/demo/table/user/' //数据接口
22. ,page: true //开启分页
23. ,cols: [[ //表头
24. {field: 'id', title: 'ID', width:80, sort: true, fixed: 'left'}
25. ,{field: 'username', title: '用户名', width:80}
26. ,{field: 'sex', title: '性别', width:80, sort: true}
27. ,{field: 'city', title: '城市', width:80}
28. ,{field: 'sign', title: '签名', width: 177}
29. ,{field: 'experience', title: '积分', width: 80, sort: true}
30. ,{field: 'score', title: '评分', width: 80, sort: true}
31. ,{field: 'classify', title: '职业', width: 80}
32. ,{field: 'wealth', title: '财富', width: 135, sort: true}
33. ]]
34. });
36. });
37. </script>
38. </body>
39. </html>

三种初始化渲染方式

|  | 方式 | 机制 | 适用场景 |
| --- | --- | --- | --- |
| 01. | [方法渲染](https://www.layui.com/doc/modules/table.html" \l "methodRender) | 用JS方法的配置完成渲染 | （推荐）无需写过多的 HTML，在 JS 中指定原始元素，再设定各项参数即可。 |
| 02. | [自动渲染](https://www.layui.com/doc/modules/table.html" \l "autoRender) | HTML配置，自动渲染 | 无需写过多 JS，可专注于 HTML 表头部分 |
| 03. | [转换静态表格](https://www.layui.com/doc/modules/table.html" \l "parseTable) | 转化一段已有的表格元素 | 无需配置数据接口，在JS中指定表格元素，并简单地给表头加上自定义属性即可 |

方法渲染

其实这是“自动化渲染”的手动模式，本质类似，只是“方法级渲染”将基础参数的设定放在了JS代码中，且原始的 table 标签只需要一个 *选择器*：

### HTML[layui.code](http://www.layui.com/doc/modules/code.html" \t "https://www.layui.com/doc/modules/_blank)

2. <table id="demo" lay-filter="test"></table>

### JavaScript[layui.code](http://www.layui.com/doc/modules/code.html" \t "https://www.layui.com/doc/modules/_blank)

1. var table = layui.table;
3. //执行渲染
4. table.render({
5. elem: '#demo' //指定原始表格元素选择器（推荐id选择器）
6. ,height: 315 //容器高度
7. ,cols: [{}] //设置表头
8. //,…… //更多参数参考右侧目录：基本参数选项
9. });

自动渲染

1. <table class="layui-table" lay-data="{height:315, url:'/demo/table/user/', page:true, id:'test'}" lay-filter="test">
2. <thead>
3. <tr>
4. <th lay-data="{field:'id', width:80, sort: true}">ID</th>
5. <th lay-data="{field:'username', width:80}">用户名</th>
6. <th lay-data="{field:'sex', width:80, sort: true}">性别</th>
7. <th lay-data="{field:'city'}">城市</th>
8. <th lay-data="{field:'sign'}">签名</th>
9. <th lay-data="{field:'experience', sort: true}">积分</th>
10. <th lay-data="{field:'score', sort: true}">评分</th>
11. <th lay-data="{field:'classify'}">职业</th>
12. <th lay-data="{field:'wealth', sort: true}">财富</th>
13. </tr>
14. </thead>
15. </table>

转换静态表格

基础参数一览表

下面是目前 table 模块所支持的全部参数一览表，我们对重点参数进行了的详细说明，你可以点击下述表格最右侧的“示例”去查看

| 参数 | 类型 | 说明 | 示例值 |
| --- | --- | --- | --- |
| elem | String/DOM | 指定原始 table 容器的选择器或 DOM，方法渲染方式必填 | "#demo" |
| cols | Array | 设置表头。值是一个二维数组。方法渲染方式必填 | [详见表头参数](https://www.layui.com/doc/modules/table.html" \l "cols) |
| url（等） | - | 异步数据接口相关参数。其中 url 参数为必填项 | [详见异步接口](https://www.layui.com/doc/modules/table.html" \l "async) |
| toolbar | String/DOM/Boolean | 开启表格头部工具栏区域，该参数支持四种类型值：   * toolbar: '#toolbarDemo' *//指向自定义工具栏模板选择器* * toolbar: '<div>xxx</div>' *//直接传入工具栏模板字符* * toolbar: true *//仅开启工具栏，不显示左侧模板* * toolbar: 'default' *//让工具栏左侧显示默认的内置模板*   注意：  1. 该参数为 layui 2.4.0 开始新增。  2. 若需要“列显示隐藏”、“导出”、“打印”等功能，则必须开启该参数 | false |
| defaultToolbar | Array | 自由配置头部工具栏右侧的图标，数组支持以下可选值：   * filter: *显示筛选图标* * exports: *显示导出图标* * print: *显示打印图标*   可根据值的顺序显示排版图标，如：  *defaultToolbar: ['filter', 'print', 'exports']*  该参数为 layui 2.4.1 新增 | ['filter', 'print'] |
| width | Number | 设定容器宽度。table容器的默认宽度是跟随它的父元素铺满，你也可以设定一个固定值，当容器中的内容超出了该宽度时，会自动出现横向滚动条。 | 1000 |
| height | Number/String | 设定容器高度 | [详见height](https://www.layui.com/doc/modules/table.html" \l "height) |
| cellMinWidth | Number | （layui 2.2.1 新增）全局定义所有常规单元格的最小宽度（默认：60），一般用于列宽自动分配的情况。其优先级低于表头参数中的 minWidth | 100 |
| done | Function | 数据渲染完的回调。你可以借此做一些其它的操作 | [详见done回调](https://www.layui.com/doc/modules/table.html" \l "done) |
| data | Array | 直接赋值数据。既适用于只展示一页数据，也非常适用于对一段已知数据进行多页展示。 | [{}, {}, {}, {}, …] |
| totalRow | Boolean | 是否开启合计行区域。layui 2.4.0 新增 | false |
| page | Boolean/Object | 开启分页（默认：false） 注：从 layui 2.2.0 开始，支持传入一个对象，里面可包含 [laypage](https://www.layui.com/doc/modules/laypage.html" \l "options) 组件所有支持的参数（jump、elem除外） | {theme: '#c00'} |
| limit | Number | 每页显示的条数（默认：10）。值务必对应 limits 参数的选项。 优先级低于 page 参数中的 limit 参数。 | 30 |
| limits | Array | 每页条数的选择项，默认：[10,20,30,40,50,60,70,80,90]。 优先级低于 page 参数中的 limits 参数。 | [30,60,90] |
| loading | Boolean | 是否显示加载条（默认：true）。如果设置 false，则在切换分页时，不会出现加载条。该参数只适用于 url 参数开启的方式 | false |
| title | String | 定义 table 的大标题（在文件导出等地方会用到）layui 2.4.0 新增 | "用户表" |
| text | Object | 自定义文本，如空数据时的异常提示等。注：layui 2.2.5 开始新增。 | [详见自定义文本](https://www.layui.com/doc/modules/table.html" \l "text) |
| autoSort | Boolean | 默认 true，即直接由 table 组件在前端自动处理排序。  若为 false，则需自主排序，通常由服务端直接返回排序好的数据。  注意：该参数为 layui 2.4.4 新增 | [详见监听排序](https://www.layui.com/doc/modules/table.html" \l "onsort) |
| initSort | Object | 初始排序状态。用于在数据表格渲染完毕时，默认按某个字段排序。 | [详见初始排序](https://www.layui.com/doc/modules/table.html" \l "initSort) |
| id | String | 设定容器唯一ID。注意：从 layui 2.2.0 开始，该参数等价于 <table id="test"></table> 中的 id 值。id值是对表格的数据操作方法上是必要的传递条件，它是表格容器的索引。你在下文也将会见识它的存在。 | test |
| skin（等） | - | 设定表格各种外观、尺寸等 | [详见表格风格](https://www.layui.com/doc/modules/table.html" \l "skin) |

cols - 表头参数一览表

| 参数 | 类型 | 说明 | 示例值 |
| --- | --- | --- | --- |
| field | String | 设定字段名。字段名的设定非常重要，且是表格数据列的唯一标识 | username |
| title | String | 设定标题名称 | 用户名 |
| width | Number/String | 设定列宽，若不填写，则自动分配；若填写，则支持值为：数字、百分比  请结合实际情况，对不同列做不同设定。 | 200 30% |
| minWidth | Number | 局部定义当前常规单元格的最小宽度（默认：60），一般用于列宽自动分配的情况。其优先级高于基础参数中的 cellMinWidth | 100 |
| type | String | 设定列类型。可选值有：   * normal（常规列，无需设定） * checkbox（复选框列） * radio（单选框列，layui 2.4.0 新增） * numbers（序号列） * space（空列） | 任意一个可选值 |
| LAY\_CHECKED | Boolean | 是否全选状态（默认：false）。必须复选框列开启后才有效，如果设置 true，则表示复选框默认全部选中。 | true |
| fixed | String | 固定列。可选值有：*left*（固定在左）、*right*（固定在右）。一旦设定，对应的列将会被固定在左或右，不随滚动条而滚动。  注意：*如果是固定在左，该列必须放在表头最前面；如果是固定在右，该列必须放在表头最后面。* | left（同 true） right |
| hide | Boolean | 是否初始隐藏列，默认：false。layui 2.4.0 新增 | true |
| totalRow | Boolean | 是否开启该列的自动合计功能，默认：false。layui 2.4.0 新增 | true |
| totalRowText | String | 用于显示自定义的合计文本。layui 2.4.0 新增 | "合计：" |
| sort | Boolean | 是否允许排序（默认：false）。如果设置 true，则在对应的表头显示排序icon，从而对列开启排序功能。  注意：*不推荐对值同时存在“数字和普通字符”的列开启排序，因为会进入字典序比对*。比如：*'贤心' > '2' > '100'*，这可能并不是你想要的结果，但字典序排列算法（ASCII码比对）就是如此。 | true |
| unresize | Boolean | 是否禁用拖拽列宽（默认：false）。默认情况下会根据列类型（type）来决定是否禁用，如复选框列，会自动禁用。而其它普通列，默认允许拖拽列宽，当然你也可以设置 true 来禁用该功能。 | false |
| edit | String | 单元格编辑类型（默认不开启）目前只支持：*text*（输入框） | text |
| event | String | 自定义单元格点击事件名，以便在 [tool](https://www.layui.com/doc/modules/table.html" \l "ontool) 事件中完成对该单元格的业务处理 | 任意字符 |
| style | String | 自定义单元格样式。即传入 CSS 样式 | background-color: #5FB878; color: #fff; |
| align | String | 单元格排列方式。可选值有：*left*（默认）、*center*（居中）、*right*（居右） | center |
| colspan | Number | 单元格所占列数（默认：1）。一般用于多级表头 | 3 |
| rowspan | Number | 单元格所占行数（默认：1）。一般用于多级表头 | 2 |
| templet | String | 自定义列模板，模板遵循 [laytpl](https://www.layui.com/doc/modules/laytpl.html" \t "https://www.layui.com/doc/modules/_blank) 语法。这是一个非常实用的功能，你可借助它实现逻辑处理，以及将原始数据转化成其它格式，如时间戳转化为日期字符等 | [详见自定义模板](https://www.layui.com/doc/modules/table.html" \l "templet) |
| toolbar | String | 绑定列工具条。设定后，可在每行列中出现一些自定义的操作性按钮 | [详见列工具条](https://www.layui.com/doc/modules/table.html" \l "toolbar) |

下面是一些方法渲染和自动渲染的配置方式：

1. //方法渲染：
2. table.render({
3. cols: [[ //标题栏
4. {checkbox: true}
5. ,{field: 'id', title: 'ID', width: 80}
6. ,{field: 'username', title: '用户名', width: 120}
7. ]]
8. });
9. 它等价于自动渲染：
10. <table class="layui-table" lay-data="{基础参数}" lay-filter="test">
11. <thead>
12. <tr>
13. <th lay-data="{checkbox:true}"></th>
14. <th lay-data="{field:'id', width:80}">ID</th>
15. <th lay-data="{field:'username', width:180}">用户名</th>
16. </tr>
17. </thead>
18. </table>

templet - 自定义列模板

**方式一：绑定模版选择器。**

1. table.render({
2. cols: [[
3. {field:'title', title: '文章标题', width: 200, templet: '#titleTpl'} //这里的templet值是模板元素的选择器
4. ,{field:'id', title:'ID', width:100}
5. ]]
6. });

下述是templet对应的模板，它可以存放在页面的任意位置。模板遵循于 [laytpl](https://www.layui.com/doc/modules/laytpl.html" \t "https://www.layui.com/doc/modules/_blank) 语法，可读取到返回的所有数据

### HTML[layui.code](http://www.layui.com/doc/modules/code.html" \t "https://www.layui.com/doc/modules/_blank)

1. <script type="text/html" id="titleTpl">
2. <a href="/detail/{{d.id}}" class="layui-table-link">{{d.title}}</a>
3. </script>
5. 注意：上述的 {{d.id}}、{{d.title}} 是动态内容，它对应数据接口返回的字段名。除此之外，你还可以读取到以下额外字段：  
    序号：{{ d.LAY\_INDEX }} （该额外字段为 layui 2.2.0 新增）
7. 由于模板遵循 laytpl 语法（建议细读 *laytpl文档* ），因此在模板中你可以写任意脚本语句（如 if else/for等）：
8. <script type="text/html" id="titleTpl">
9. {{# if(d.id < 100){ }}
10. <a href="/detail/{{d.id}}" class="layui-table-link">{{d.title}}</a>
11. {{# } else { }}
12. {{d.title}}(普通用户)
13. {{# } }}
14. </script>

**方式二：函数转义**

### code[layui.code](http://www.layui.com/doc/modules/code.html" \t "https://www.layui.com/doc/modules/_blank)

1. table.render({
2. cols: [[
3. {field:'title', title: '文章标题', width: 200
4. ,templet: function(d){
5. return 'ID：'+ d.id +'，标题：<span style="color: #c00;">'+ d.title +'</span>'
6. }
7. }
8. ,{field:'id', title:'ID', width:100}
9. ]]
10. });

**方式三：直接赋值模版字符**

1. templet: '<div><a href="/detail/{{d.id}}" class="layui-table-link">{{d.title}}</a></div>'

toolbar - 绑定列工具条

1. table.render({
2. cols: [[
3. {field:'id', title:'ID', width:100}
4. ,{fixed: 'right', width:150, align:'center', toolbar: '#barDemo'} //这里的toolbar值是模板元素的选择器
5. ]]
6. });

下述是 toolbar 对应的模板，它可以存放在页面的任意位置：

### HTML[layui.code](http://www.layui.com/doc/modules/code.html" \t "https://www.layui.com/doc/modules/_blank)

1. <script type="text/html" id="barDemo">
2. <a class="layui-btn layui-btn-xs" lay-event="detail">查看</a>
3. <a class="layui-btn layui-btn-xs" lay-event="edit">编辑</a>
4. <a class="layui-btn layui-btn-danger layui-btn-xs" lay-event="del">删除</a>
6. <!-- 这里同样支持 laytpl 语法，如： -->
7. {{# if(d.auth > 2){ }}
8. <a class="layui-btn layui-btn-xs" lay-event="check">审核</a>
9. {{# } }}
10. </script>

异步数据接口

done - 数据渲染完的回调

类型：*Function*，默认值：*无*

无论是异步请求数据，还是直接赋值数据，都会触发该回调。

1. table.render({ //其它参数在此省略
2. done: function(res, curr, count){
3. //如果是异步请求数据方式，res即为你接口返回的信息。
4. //如果是直接赋值的方式，res即为：{data: [], count: 99} data为当前页数据、count为数据总长度
5. console.log(res);
6. //得到当前页码
7. console.log(curr);
8. //得到数据总量
9. console.log(count);
10. }
11. });

initSort - 初始排序

1. //“方法级渲染”配置方式
2. table.render({ //其它参数在此省略
3. initSort: {
4. field: 'id' //排序字段，对应 cols 设定的各字段名
5. ,type: 'desc' //排序方式 asc: 升序、desc: 降序、null: 默认排序
6. }
7. });

height - 设定容器高度

height - 设定容器高度

类型：*Number/String*，可选值如下：

| 可选值 | 说明 | 示例 |
| --- | --- | --- |
| 不填写 | 默认情况。高度随数据列表而适应，表格容器不会出现纵向滚动条 | - |
| 固定值 | 设定一个数字，用于定义容器高度，当容器中的内容超出了该高度时，会自动出现纵向滚动条 | height: 315 |
| full-差值 | 高度将始终铺满，无论浏览器尺寸如何。这是一个特定的语法格式，其中 *full* 是固定的，而 *差值*则是一个数值，这需要你来预估，比如：表格容器距离浏览器顶部和底部的距离“和”  PS：*该功能为 layui 2.1.0 版本中新增* | height: 'full-20' |

获取选中行

该方法可获取到表格所有的选中行相关数据   
语法：*table.checkStatus('idTest')* idTest为基础参数id对应的值（见：[设定容器唯一ID](https://www.layui.com/doc/modules/table.html" \l "id)），如：

### 渲染方式[layui.code](http://www.layui.com/doc/modules/code.html" \t "https://www.layui.com/doc/modules/_blank)

1. 【自动化渲染】
2. <table class="layui-table" lay-data="{id: 'idTest'}"> …… </table>
4. 【方法渲染】
5. table.render({ //其它参数省略
6. id: 'idTest'
7. });

监听头部工具栏事件

点击头部工具栏区域设定了属性为 *lay-event=""* 的元素时触发（该事件为 layui 2.4.0 开始新增）。

1. 原始容器
2. <table id="demo" lay-filter="test"></table>
4. 工具栏模板：
5. <script type="text/html" id="toolbarDemo">
6. <div class="layui-btn-container">
7. <button class="layui-btn layui-btn-sm" lay-event="add">添加</button>
8. <button class="layui-btn layui-btn-sm" lay-event="delete">删除</button>
9. <button class="layui-btn layui-btn-sm" lay-event="update">编辑</button>
10. </div>
11. </script>
13. <script;>
14. //JS 调用：
15. table.render({
16. elem: '#demo'
17. ,toolbar: '#toolbarDemo'
18. //,…… //其他参数
19. });
21. //监听事件
22. table.on('toolbar(test)', function(obj){
23. var checkStatus = table.checkStatus(obj.config.id);
24. switch(obj.event){
25. case 'add':
26. layer.msg('添加');
27. break;
28. case 'delete':
29. layer.msg('删除');
30. break;
31. case 'update':
32. layer.msg('编辑');
33. break;
34. };
35. });
36. </script>

监听单元格编辑

1. table.on('edit(test)', function(obj){ //注：edit是固定事件名，test是table原始容器的属性 lay-filter="对应的值"
2. console.log(obj.value); //得到修改后的值
3. console.log(obj.field); //当前编辑的字段名
4. console.log(obj.data); //所在行的所有相关数据
5. });

表格重载

很多时候，你需要对表格进行重载。比如数据全局搜索。以下方法可以帮你轻松实现这类需求（可任选一种）。

| 语法 | 说明 | 适用场景 |
| --- | --- | --- |
| table.reload(ID, options) | 参数 *ID* 即为基础参数id对应的值，见：[设定容器唯一ID](https://www.layui.com/doc/modules/table.html" \l "id)  参数 *options* 即为各项基础参数  注意：*该方法为 2.1.0 版本中新增* | 所有渲染方式 |
| tableIns.reload(options) | 对象 *tableIns* 来源于 table.render() 方法的实例  参数 *options* 即为各项基础参数 | 仅限方法级渲染 |