快速使用

laypage 的使用非常简单，指向一个用于存放分页的容器，通过服务端得到一些初始值，即可完成分页渲染：

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="utf-8">
5. <title>layPage快速使用</title>
6. <link rel="stylesheet" href="/static/build/layui.css" media="all">
7. </head>
8. <body>
9. <div id="test1"></div>
10. <script src="/static/build/layui.js"></script>
11. <script>
12. layui.use('laypage', function(){
13. var laypage = layui.laypage;
14. //执行一个laypage实例
15. laypage.render({
16. elem: 'test1' //注意，这里的 test1 是 ID，不用加 # 号
17. ,count: 50 //数据总数，从服务端得到
18. });
19. });
20. </script>
21. </body>
22. </html>

基础参数选项

通过核心方法：*laypage.render(options)* 来设置基础参数。由于使用非常简单，本篇直接罗列核心接口的参数选项

| 参数选项 | 说明 | 类型 | 默认值 |
| --- | --- | --- | --- |
| elem | 指向存放分页的容器，值可以是容器ID、DOM对象。如：  1. elem: 'id' *注意：这里不能加 # 号*  2. elem: document.getElementById('id') | String/Object | - |
| count | 数据总数。一般通过服务端得到 | Number | - |
| limit | 每页显示的条数。laypage将会借助 count 和 limit 计算出分页数。 | Number | 10 |
| limits | 每页条数的选择项。如果 layout 参数开启了 limit，则会出现每页条数的select选择框 | Array | [10, 20, 30, 40, 50] |
| curr | 起始页。一般用于刷新类型的跳页以及HASH跳页。如： code[layui.code](http://www.layui.com/doc/modules/code.html" \t "https://www.layui.com/doc/modules/_blank)  1. //开启location.hash的记录 2. laypage.render({ 3. elem: 'test1' 4. ,count: 500 5. ,curr: location.hash.replace('#!fenye=', '') //获取起始页 6. ,hash: 'fenye' //自定义hash值 7. }); | Number | 1 |
| groups | 连续出现的页码个数 | Number | 5 |
| prev | 自定义“上一页”的内容，支持传入普通文本和HTML | String | 上一页 |
| next | 自定义“下一页”的内容，同上 | String | 下一页 |
| first | 自定义“首页”的内容，同上 | String | 1 |
| last | 自定义“尾页”的内容，同上 | String | 总页数值 |
| layout | 自定义排版。可选值有：*count*（总条目输区域）、*prev*（上一页区域）、*page*（分页区域）、*next*（下一页区域）、*limit*（条目选项区域）、*refresh*（页面刷新区域。注意：layui 2.3.0 新增） 、*skip*（快捷跳页区域） | Array | ['prev', 'page', 'next'] |
| theme | 自定义主题。支持传入：*颜色值*，或*任意普通字符*。如：  1. theme: '#c00'  2. theme: 'xxx' //将会生成 class="layui-laypage-xxx" 的CSS类，以便自定义主题 | String | - |
| hash | 开启location.hash，并自定义 hash 值。如果开启，在触发分页时，会自动对url追加：*#!hash值={curr}* 利用这个，可以在页面载入时就定位到指定页 | String/Boolean | false |

jump - 切换分页的回调

当分页被切换时触发，函数返回两个参数：*obj*（当前分页的所有选项值）、first（是否首次，一般用于初始加载的判断）

1. laypage.render({
2. elem: 'test1'
3. ,count: 70 //数据总数，从服务端得到
4. ,jump: function(obj, first){
5. //obj包含了当前分页的所有参数，比如：
6. console.log(obj.curr); //得到当前页，以便向服务端请求对应页的数据。
7. console.log(obj.limit); //得到每页显示的条数
9. //首次不执行
10. if(!first){
11. //do something
12. }
13. }
14. });