

自定义组件

参考

<https://element.eleme.cn/#/zh-CN/component/button>

封装自己的：按钮组件

封装原因

按钮在后台的程序开发中，大量的存在，每个页面几乎都会使用按钮，所以按钮特别适合做通用组件。

实现步骤

- 在封装之前，一般你们公司UI会设计一套完整的按钮原型
- 前端会进行切图，进行html化
- 然后进行组件封装（bootstrap css封装,vue进行组件按钮封装）

分组分页组件

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport"
6         content="width=device-width, user-
scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
7     <meta http-equiv="X-UA-Compatible"
content="ie=edge">
8     <title>21、vue的组件自定义按钮组件-默认elementui-
05.html</title>
9     <link rel="stylesheet" href="css/ksdui.css">
10    <link rel="stylesheet"
href="//at.alicdn.com/t/font_2882935_q0cj641ukuh
.css">
11 </head>
12 <body>
13
14
15 <div id="app">
16
17     <ksd-page
18         :page-index="currentPage"
19         :total="count"
20         :is-go="false"
21         :is-total="false"
22         :page-size="pageSize"
23         @change="pageChange">
```

```
24     </ksd-page>
25
26     <ksd-page
27         :page-index="currentPage"
28         :total="count"
29         :is-go="false"
30         :is-total="true"
31         :page-size="pageSize"
32         @change="pageChange">
33     </ksd-page>
34 </div>
35
36 <script src="js/vue.global.js"></script>
37 <script>
38
39     // 1: 根组件
40     var app = Vue.createApp({
41         data() {
42             return {
43                 count: 100,
44                 pageSize: 10, //每页显示20条数据
45                 currentPage: 1, //当前页码
46                 items: []
47             }
48         },
49         created() {
50
51         },
52         methods: {
53             pageChange(pageNo) {
54
```

```
55         }
56     }
57 });
58 // 2: 子组件 -- 必须放在父组件内部
59 app.component("ksd-page", {
60     props: {
61         //页面中的可见页码，其他的以...替代，必须
        是奇数
62         perPages: {
63             type: Number,
64             default: 5
65         },
66
67         //当前页码
68         pageIndex: {
69             type: Number,
70             default: 1
71         },
72
73         //每页显示条数
74         pageSize: {
75             type: Number,
76             default: 10
77         },
78
79         //总记录数
80         total: {
81             type: Number,
82             default: 1
83         },
84     },
```

```
85         isGo:{
86             type:Boolean,
87             default:false
88         },
89
90         isTotal:{
91             type:Boolean,
92             default:false
93         }
94     },
95     template: "<div><ul class=\"ksd-
paging\">\n" +
96         "        <!-- prev -->\n" +
97         "        <li\n" +
98         "            :class=\"['paging-item',
'paging-item--prev', {'paging-item--disabled' :
index == 1}]\n" +
99         "            @click=\"prev\">上一页
</li>\n" +
100         "\n" +
101         "        <!-- first -->\n" +
102         "        <li\n" +
103         "            :class=\"['paging-item',
'paging-item--first', {'paging-item--disabled' :
index == 1}]\n" +
104         "            @click=\"first\">首页
</li>\n" +
105         "\n" +
106         "        <li\n" +
107         "            :class=\"['paging-item',
'paging-item--more']\n" +
```

```

108         "                v-if=\"showPrevMore\">...
</li>\n" +
109         "\n" +
110         "        <li\n" +
111         "            :class=\"['paging-item',
{'paging-item--current' : index == pager}]\n"
+
112         "                v-for=\"pager in
pagers\n" +
113         "                @click=\"go(pager)\n"
{{pager}}</li>\n" +
114         "\n" +
115         "        <li\n" +
116         "            :class=\"['paging-item',
'paging-item--more']\n" +
117         "                v-if=\"showNextMore\">...
</li>\n" +
118         "\n" +
119         "        <!-- last -->\n" +
120         "        <li\n" +
121         "            :class=\"['paging-item',
'paging-item--last', {'paging-item--disabled' :
index == pages}]\n" +
122         "                @click=\"last\n" +
+
123         "\n" +
124         "        <!-- next -->\n" +
125         "        <li\n" +
126         "            :class=\"['paging-item',
'paging-item--next', {'paging-item--disabled' :
index == pages}]\n" +

```

```
127         "          @click=\"next\">下一页
</li>\n" +
128         "          <li v-if='isGo' :class=\"
['paging-item', 'paging-item--next']\"><input
class='pageinput' type='text' v-model='index'>
</li>\n" +
129         "          <li v-if='isTotal'
:class=\"['paging-item', 'paging-item--next']\">
共{{total}}条</li>\n" +
130         "    </ul></div>",
131
132     methods: {
133       toPage(){
134         this.go(this.index)
135       },
136       prev() {
137         if (this.index > 1) {
138           this.go(this.index - 1)
139         }
140       },
141       next() {
142         if (this.index < this.pages) {
143           this.go(this.index + 1)
144         }
145       },
146       first() {
147         if (this.index !== 1) {
148           this.go(1)
149         }
150       },
151       last() {
```

```
152         if (this.index !== this.pages) {
153             this.go(this.pages)
154         }
155     },
156     go(page) {
157         if (this.index !== page) {
158             this.index = page
159             //父组件通过change方法来接受当前
    的页码
160             this.$emit('change',
    this.index)
161         }
162     },
163 },
164
165     computed: {
166
167         //计算总页码
168         pages() {
169             return Math.ceil(this.size /
    this.limit)
170         },
171
172         //计算页码，当count等变化时自动计算
173         pagers() {
174             const array = []
175             const perPages = this.perPages *
    1
176             const pageCount = this.pages * 1
177             let current = this.index * 1
```



```
178         const _offset = (perPages - 1) /
179         2
180         const offset = {
181             start: current - _offset,
182             end: current + _offset
183         }
184         //-1, 3
185         if (offset.start < 1) {
186             offset.end = offset.end + (1
187             - offset.start)
188             offset.start = 1
189         }
190         if (offset.end > pageCount) {
191             offset.start = offset.start
192             - (offset.end - pageCount)
193             offset.end = pageCount
194         }
195         if (offset.start < 1)
196         offset.start = 1
197
198         this.showPrevMore =
199         (offset.start > 1)
200         this.showNextMore = (offset.end
201         < pageCount)
```

```

202         return array
203     }
204 },
205
206     data() {
207         return {
208             index: this.pageIndex, //当前页码
209             limit: this.pageSize, //每页显示条
210             size: this.total || 1, //总记录数
211             showPrevMore: false,
212             showNextMore: false
213         }
214     },
215     watch: {
216         pageIndex(val) {
217             this.index = val || 1
218         },
219         pageSize(val) {
220             this.limit = val || 10
221         },
222         total(val) {
223             this.size = val || 1
224         }
225     }
226 });
227
228 // 3:vue视图挂载
229 app.mount("#app");
230
231 </script>

```

232

233 </body>

234 </html>