Layui 宇宙版教程 version1.6

《Layui 宇宙版教程》提供 2000 人的 QQ 群进行交流学习，QQ 群号：1046961650，或通过手机 QQ 扫描二维码进入：

任职公司中大部分的软件后台都在使用 Layui，的确像 Layui 作者“贤心”所说：

该框架是专为后端服务员量身定做，无需接触过多的前端技术即可完成后台界面的开发，快速实现业务逻辑，而且界面美观，组件丰富。

十分赞成贤心的总结，而且我还相信一句话：不管是什么技术，提高生产力才是有力量的！

学习 Layui 的成本非常低，以致于看看官方的帮助文档都可以快速上手搭建商业级的 UI 软件界面，但在学习 Layui 的途中却又布满荆棘，有时一个小小的属性却要研究个把小时来琢磨它的作用与显示的效果，加之官方文档也不可能面面俱到，有些技术点还需要自己趟坑解决，为了让学习 Layui 的初学者有更快的学习效率，所以催生了“Layui 宇宙版教程”的诞生！

请原谅我用“宇宙版”这 3 个字~！但它的确真的很全！600 多页的文档已经把 Layui 大部分常用组件的使用进行了覆盖式讲解，力求详细，案例能直接使用。

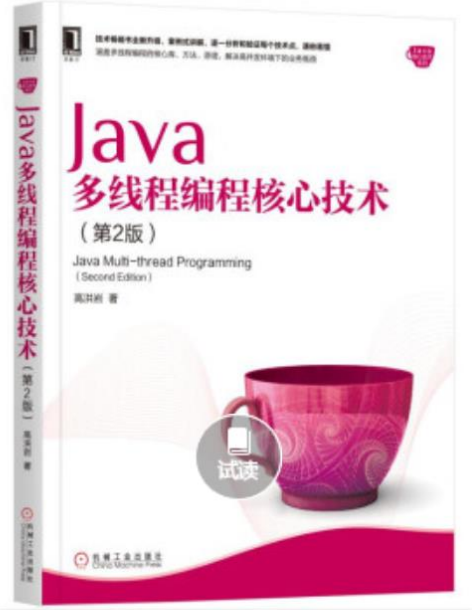
本教程是根据 http://www.layui.com 官方在线文档进行整理，每个案例都可运行，基本属于 CV 大法即可快速跑起来，无需自己写代码，让读者快速掌握 Layui 的使用，但在整理过程中有可能出现遗漏或差错，请各位指正。

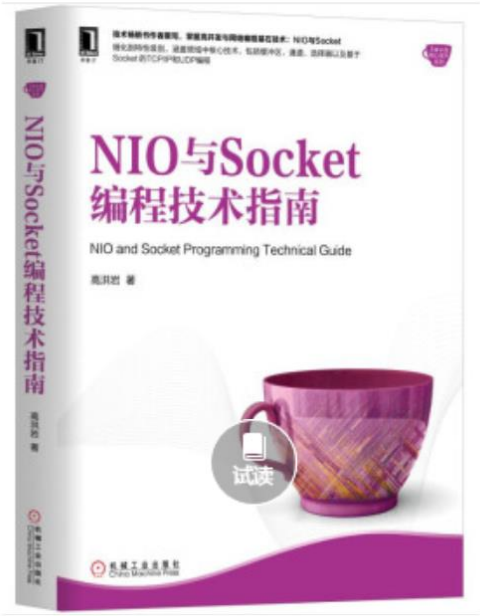
本教程作者：**高洪岩**，居住地北京，他是一个热爱技术，愿意分享技术并且爱好摄影的程序员，使用器材：

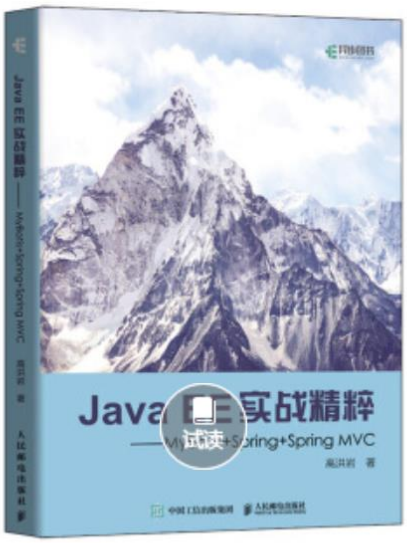
EOS-1DX Mark II
EF 50mm f/1.2L USM
EF 85mm f/1.2L II USM
EF 135mm f/2L USM
EF 200mm f/2L IS USM
TS-E 17mm f/4L
EF 24-70mm f/2.8L II USM
EOS R

RF 35mm F1.8 MACRO IS STM
Canon 600EX II-RT
方片滤镜套装及 17 移轴滤镜支架

作者相关已出版书籍如图 1-xx 所示。







1.1 Layui 介绍

Layui 是经典的模块化前端框架，由职业前端资深工程师倾情打造，面向全层次的前后端开发者和低门槛开箱即用的前端 UI 解决方案。

Layui 是采用模块化规范编写的前端 UI 框架，遵循原生 HTML/CSS/JS 的书写与组织形式，学习门槛极低，拿来即用。其外在极简，却又不失饱满的内在，体积轻巧，组件丰富，从核心代码到 API 的每一处细节都经过精心雕琢，非常适合界面的快速开发。

Layui 首个版本发布于 2016 年金秋，她更多是为服务端程序员量身定做，无需涉足各种前端工具的复杂配置，只需面对浏览器本身，实现快速开发。

Layui 兼容人类正在使用的全部浏览器（IE6/7 除外），可作为 PC 端后台系统与前台界面的速成开发方案。

Layui 的 Logo 如图 1-xx 所示。

图 1-xx

1.2 搭建 Layui 开发环境与两种使用方式

搭建 Layui 开发环境与两种使用方式。

1.2.1 搭建 Layui 开发环境

Layui 官方网址如下：
https://www.layui.com/
当前最新版本为 2.5.6，如图 1-xx 所示。

图 1-xx

下载量已经达到 1653422 次，足以见证 Layui 的流行。

Layui 是开源的项目，github 和 gitee 地址如下：
https://github.com/sentsin/layui/
https://gitee.com/sentsin/layui

Layui 官方公众号如图 1-xx 所示。

图 1-xx

在 Layui 官网中点击“立即下载”按钮开始下载 Layui，下载文件是 layui-v2.5.6.zip。解压 layui-v2.5.6.zip 文件，内容如图 1-xx 所示。

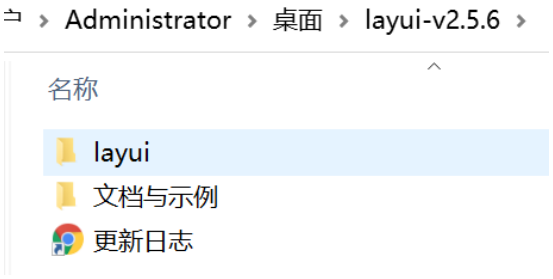


图 1-xx

进入“layui”文件夹，如图 1-xx 所示。

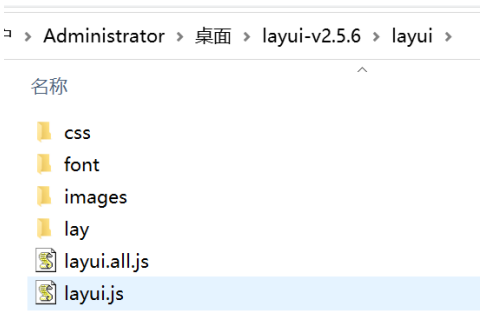


图 1-xx

继续进入“css”文件夹，如图 1-xx 所示。

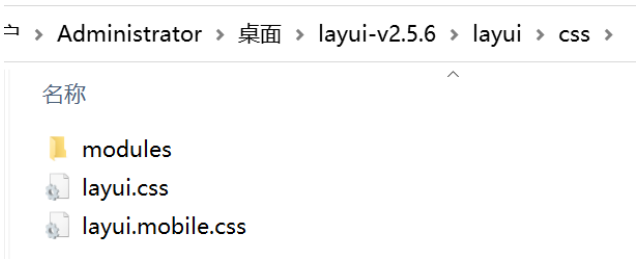


图 1-xx

使用 Layui 开发时，需要引入 layui.all.js 和 layui.css 文件。

整体目录结构如图 1-xx 所示。

```
├─css //css目录
|   |─modules //模块css目录（一般如果模块相对较大，我们会单独提取，比如下面三个：）
|   |   ├─laydate
|   |   ├─layer
|   |   └─layim
|   └─layui.css //核心样式文件
├─font  //字体图标目录
├─images //图片资源目录（目前只有layim和编辑器用到的GIF表情）
|─lay //模块核心目录
|   └─modules //各模块组件
|─layui.js //基础核心库
└─layui.all.js //包含layui.js和所有模块的合并文件
```

图 1-xx

进入“文档与示例”文件夹，如图 1-xx 所示。

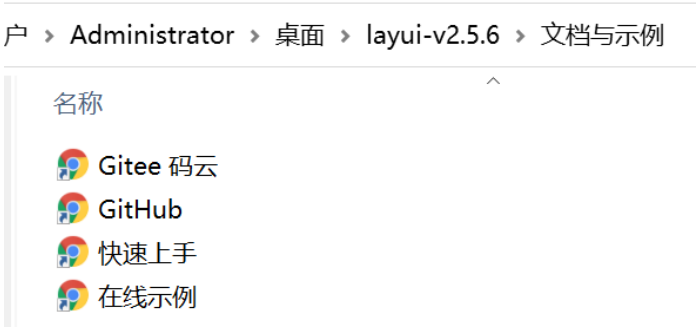


图 1-xx

由于 Layui 是中国人开发的，所以开发文档非常友好，易于学习。

1.2.2 两种使用方式

两种使用方式：

（1）模块化：加载指定模块。

（2）非模块化：加载全部模块。

使用 Layui 需要引入 layui/css/layui.css 和 layui/layui.js 文件。如果是采用非模块化方式可以引入 layui/layui.all.js 文件。

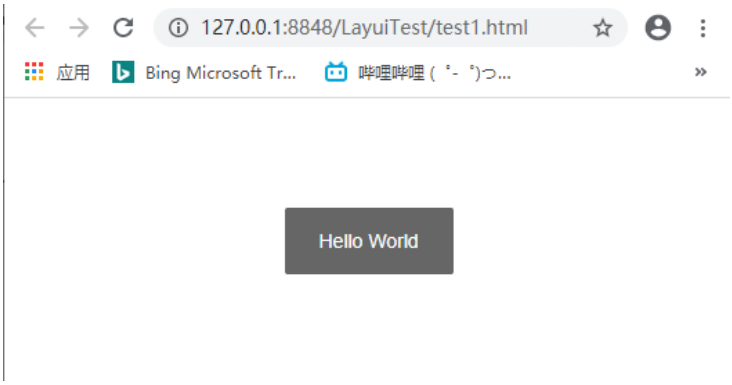
1.2.2.1 加载单独模块

示例代码如下：

```
<!DOCTYPE html>
```

```
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
    </head>
    <body>
        <script src="layui/layui.js"></script>
        <script>
            //一般直接写在一个js文件中
            layui.use(['layer', 'form'], function() {
                var layer = layui.layer;
                var form = layui.form;
                layer.msg('Hello World');
            });
        </script>
    </body>
</html>
```

针对性的加载了 layer 和 form 模块。
运行结果如图 1-xx 所示。



为了提高运行效率，使用如下代码加载指定的模块：

```
            layui.use(['layer', 'form'], function() {
                var layer = layui.layer;
                var form = layui.form;
                layer.msg('Hello World');
            });
```

1.2.2.2 加载全部模块

加载全部模块的示例代码如下：

```
<!DOCTYPE html>
```

```
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            var form = layui.form;
            layer.msg('Hello World');
        </script>
    </body>
</html>
```

不需要使用 layui.use()方法加载指定模块，因为全部模块已经自动加载完毕。

如果是线上环境，更推荐使用全模块加载，即直接引入 layui.all.js 文件，它包含了 layui 所有的内置模块，无需再通过 layui.use()方法加载模块，直接调用即可。

1.3 布局

在 Layui2.0 的版本中加入了强劲的栅格系统和后台布局方案，这意味着终于可以着手采用 Layui 排版响应式网站和后台系统了。

Layui 的栅格系统采用业界比较常见的 12 等分规则，内置：

（1）移动设备

（2）平板

（3）桌面中等屏幕

（4）大型屏幕

多终端适配处理，最低能支持到 IE8。

1.3.1 栅格系统

为了丰富网页布局，简化 HTML/CSS 代码的耦合，并提升多终端的适配能力，Layui 在 2.0 的版本中引进了自己的一套具备响应式能力的栅格系统。对容器进行了 12 等分，预设了 4*12 种 CSS 排版类，它们在移动设备、平板、桌面中等屏幕，大型屏幕这四种不同大小规格的屏幕下发挥着各自的作用。

使用栅格系统需要借助于 class="layui-row"样式来定义行，示例代码如下：

使用栅格系统需要借助于 class="layui-col-md*"样式来定义列，示例代码如下：

列 要放在行 内部。

变量 md 代表的是不同屏幕大小下的标记。可以取值 xs（超小屏幕，如手机），sm（小屏幕，如平板），md（桌面中等屏幕），lg（桌面大型屏幕）。

变量*代表的是该列占用 12 等分，和容器的宽度一样。可选值为 1-12。

如果一行中多个列的“等分数”总和等于 12，则刚好满行排列。如果大于 12，多余的列将自动另起一行进行显示。

响应式规则如图 1-xx 所示。

栅格的响应式能力，得益于CSS3媒体查询（Media Queries）的强力支持，从而针对四类不同尺寸的屏幕，进行相应的适配处理

	超小屏幕 (手机<768px)	小屏幕 (平板≥768px)	中等屏幕 (桌面≥992px)	大型屏幕 (桌面≥1200px)
.layui-container 的值	auto	750px	970px	1170px
标记	xs	sm	md	lg
列对应类 * 为1-12的等分数值	layui-col-xs*	layui-col-sm*	layui-col-md*	layui-col-lg*
总列数	12			
响应行为	始终按设定的比例水平排列	在当前屏幕下水平排列，如果屏幕大小低于临界值则堆叠排列		

图 1-xx

1.3.2 测试 xs

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-xs1" style="background-color:
red;">1</div><br />
            <div class="layui-col-xs2" style="background-color:
red;">2</div><br />
            <div class="layui-col-xs3" style="background-color:
red;">3</div><br />
            <div class="layui-col-xs4" style="background-color:
red;">4</div><br />
            <div class="layui-col-xs5" style="background-color:
red;">5</div><br />
            <div class="layui-col-xs6" style="background-color:
```

```
red;">6</div><br />
            <div class="layui-col-xs7" style="background-color:
red;">7</div><br />
            <div class="layui-col-xs8" style="background-color:
red;">8</div><br />
            <div class="layui-col-xs9" style="background-color:
red;">9</div><br />
            <div class="layui-col-xs10" style="background-color:
red;">10</div><br />
            <div class="layui-col-xs11" style="background-color:
red;">11</div><br />
            <div class="layui-col-xs12" style="background-color:
red;">12</div><br /><br />
            <div class="layui-col-xs*" style="background-color:
red;">12</div><br />
        </div>
    </body>
</html>
```

运行结果如图 1-xx 所示。

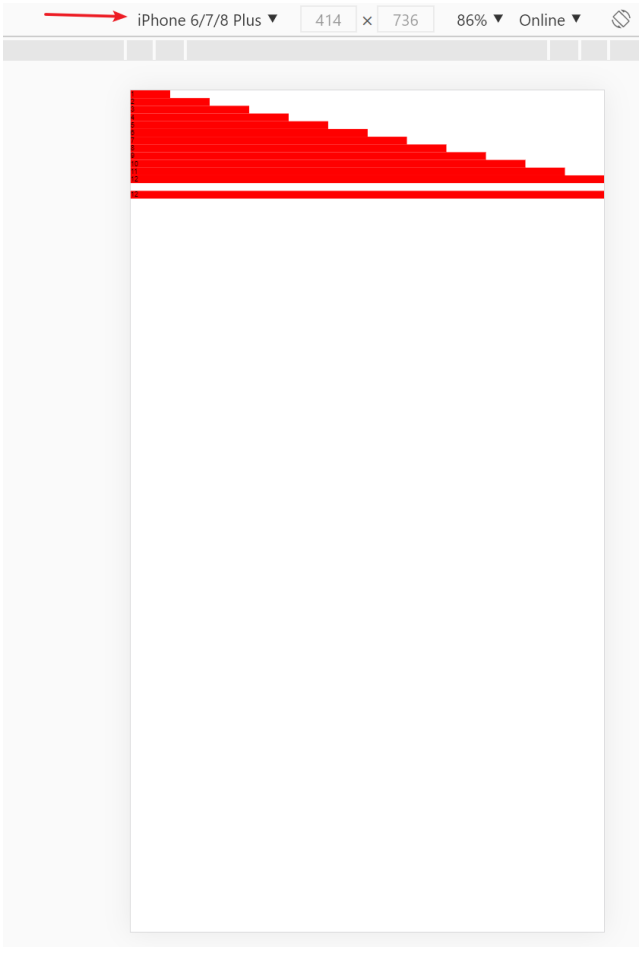


图 1-xx

1.3.3 测试 sm

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-sm1" style="background-color:
red;">1</div><br />
            <div class="layui-col-sm2" style="background-color:
red;">2</div><br />
            <div class="layui-col-sm3" style="background-color:
red;">3</div><br />
            <div class="layui-col-sm4" style="background-color:
red;">4</div><br />
            <div class="layui-col-sm5" style="background-color:
red;">5</div><br />
            <div class="layui-col-sm6" style="background-color:
red;">6</div><br />
            <div class="layui-col-sm7" style="background-color:
red;">7</div><br />
            <div class="layui-col-sm8" style="background-color:
red;">8</div><br />
            <div class="layui-col-sm9" style="background-color:
red;">9</div><br />
            <div class="layui-col-sm10" style="background-color:
red;">10</div><br />
            <div class="layui-col-sm11" style="background-color:
red;">11</div><br />
            <div class="layui-col-sm12" style="background-color:
red;">12</div><br /><br />
            <div class="layui-col-sm*" style="background-color:
red;">12</div><br />
        </div>
    </body>
</html>
```

运行结果如图 1-xx 所示。

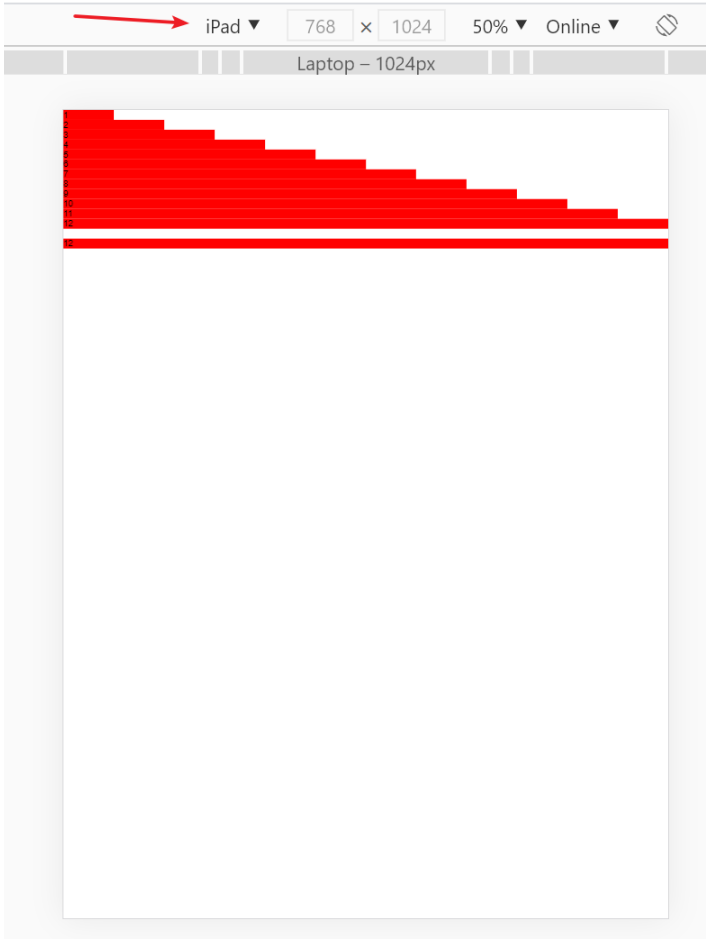


图 1-xx

1.3.4 测试 md

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-md1" style="background-color:
red;">1</div><br />
            <div class="layui-col-md2" style="background-color:
```

```
red;">2</div><br />
            <div class="layui-col-md3" style="background-color:
red;">3</div><br />
            <div class="layui-col-md4" style="background-color:
red;">4</div><br />
            <div class="layui-col-md5" style="background-color:
red;">5</div><br />
            <div class="layui-col-md6" style="background-color:
red;">6</div><br />
            <div class="layui-col-md7" style="background-color:
red;">7</div><br />
            <div class="layui-col-md8" style="background-color:
red;">8</div><br />
            <div class="layui-col-md9" style="background-color:
red;">9</div><br />
            <div class="layui-col-md10" style="background-color:
red;">10</div><br />
            <div class="layui-col-md11" style="background-color:
red;">11</div><br />
            <div class="layui-col-md12" style="background-color:
red;">12</div><br /><br />
            <div class="layui-col-md*" style="background-color:
red;">12</div><br />
        </div>
    </body>
</html>
```

运行结果如图 1-xx 所示。

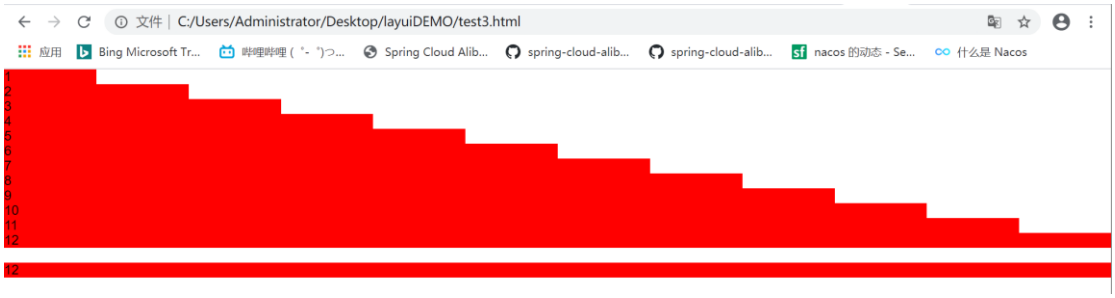


图 1-xx

1.3.5 测试 lg

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
```

```
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-lg1" style="background-color:
red;">1</div><br />
            <div class="layui-col-lg2" style="background-color:
red;">2</div><br />
            <div class="layui-col-lg3" style="background-color:
red;">3</div><br />
            <div class="layui-col-lg4" style="background-color:
red;">4</div><br />
            <div class="layui-col-lg5" style="background-color:
red;">5</div><br />
            <div class="layui-col-lg6" style="background-color:
red;">6</div><br />
            <div class="layui-col-lg7" style="background-color:
red;">7</div><br />
            <div class="layui-col-lg8" style="background-color:
red;">8</div><br />
            <div class="layui-col-lg9" style="background-color:
red;">9</div><br />
            <div class="layui-col-lg10" style="background-color:
red;">10</div><br />
            <div class="layui-col-lg11" style="background-color:
red;">11</div><br />
            <div class="layui-col-lg12" style="background-color:
red;">12</div><br /><br />
            <div class="layui-col-lg*" style="background-color:
red;">12</div><br />
        </div>
    </body>
</html>
```

运行结果如图 1-xx 所示。

图 1-xx

1.3.6 实现不同列宽

xs，sm，md，lg 这四者运行效果看似差不多，区别是当浏览器宽度缩小时出现响应式的时机不一样。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-lg2" style="background-color:
red;">2</div>
            <div class="layui-col-lg3" style="background-color:
green;">3</div>
            <div class="layui-col-lg7" style="background-color:
blue;">7</div>
        </div>
        <span id="screenWidth"></span>
        <script>
            $(window).resize(function() {
                $("#screenWidth").text($(window).width());
            });
        </script>
    </body>
</html>
```

当浏览器宽度>=1200 像素时运行结果如图 1-xx 所示。

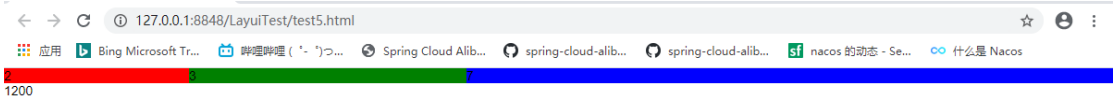


图 1-xx

当浏览器宽度<1200 像素时运行结果如图 1-xx 所示。

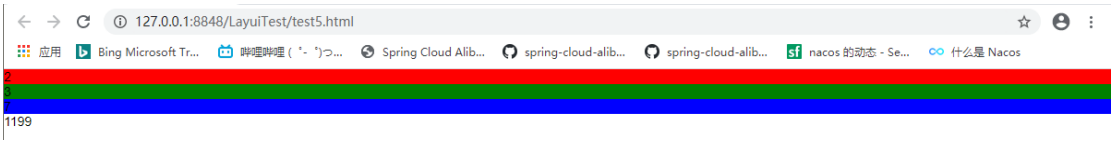


图 1-xx

以 1200 像素为分割点的原因是 lg 规则定义的，如图 1-xx 所示。

栅格的响应式能力，得益于CSS3媒体查询（Media Queries）的强力支持，从而针对四类不同尺寸的屏幕，进行相应的适配处理

	超小屏幕 (手机<768px)	小屏幕 (平板≥768px)	中等屏幕 (桌面≥992px)	大型屏幕 (桌面≥1200px)
.layui-container 的值	auto	750px	970px	1170px
标记	xs	sm	md	lg
列对应类 * 为1-12的等分数值	layui-col-xs*	layui-col-sm*	layui-col-md*	layui-col-lg*
总列数	12			
响应行为	始终按设定的比例水平排列	在当前屏幕下水平排列，如果屏幕大小低于临界值则堆叠排列		

图 1-xx

请自行测试 md 以 992 像素为分割点的效果。
请自行测试 sm 以 768 像素为分割点的效果。

1.3.7 测试移动设备、平板、桌面端的不同表现

测试移动设备、平板、桌面端的不同表现。

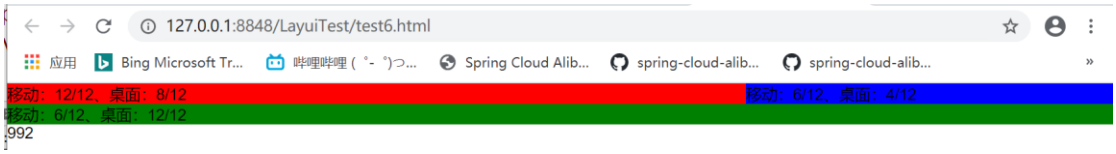
1.3.7.1 测试 1

测试代码如下：

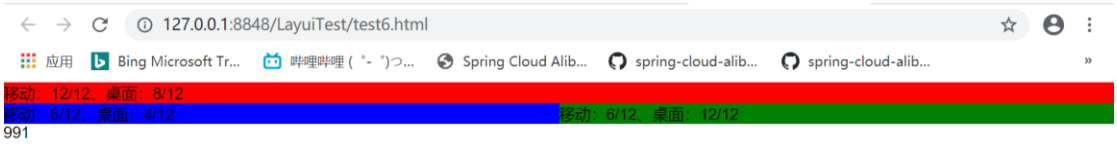
```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-xs12 layui-col-md8">
                <div style="background-color: red;">移动：12/12、桌面：
8/12</div>
            </div>
```

```
            <div class="layui-col-xs6 layui-col-md4">
                <div style="background-color: blue;">移动：6/12、桌面：
4/12</div>
            </div>
            <div class="layui-col-xs6 layui-col-md12">
                <div style="background-color: green;">移动：6/12、桌面：
12/12</div>
            </div>
        </div>
        <span id="screenWidth"></span>
        <script>
            $(window).resize(function() {
                $("#screenWidth").text($(window).width());
            });
        </script>
    </body>
</html>
```

中等屏幕 md 中运行结果如图 1-xx 所示。



手机 xs 中运行结果如图 1-xx 所示。



1.3.7.2 测试 2

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
```

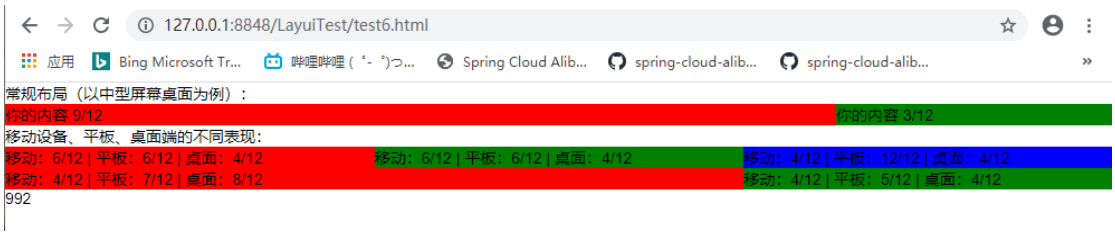
常规布局（以中型屏幕桌面为例）：

```
<div class="layui-row">
    <div class="layui-col-md9" style="background-color: red;">
        你的内容 9/12
    </div>
    <div class="layui-col-md3" style="background-color: green;">
        你的内容 3/12
    </div>
</div>
```

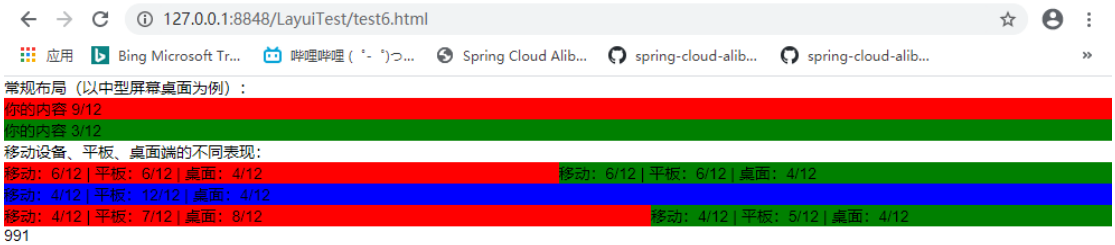
移动设备、平板、桌面端的不同表现：

```
<div class="layui-row">
    <div class="layui-col-xs6 layui-col-sm6 layui-col-md4"
style="background-color: red;">
        移动：6/12 | 平板：6/12 | 桌面：4/12
    </div>
    <div class="layui-col-xs6 layui-col-sm6 layui-col-md4"
style="background-color: green;">
        移动：6/12 | 平板：6/12 | 桌面：4/12
    </div>
    <div class="layui-col-xs4 layui-col-sm12 layui-col-md4"
style="background-color: blue;">
        移动：4/12 | 平板：12/12 | 桌面：4/12
    </div>
    <div class="layui-col-xs4 layui-col-sm7 layui-col-md8"
style="background-color: red;">
        移动：4/12 | 平板：7/12 | 桌面：8/12
    </div>
    <div class="layui-col-xs4 layui-col-sm5 layui-col-md4"
style="background-color: green;">
        移动：4/12 | 平板：5/12 | 桌面：4/12
    </div>
</div>
<span id="screenWidth"></span>
<script>
    $(window).resize(function() {
        $("#screenWidth").text($(window).width());
    });
</script>
</body>
</html>
```

中等屏幕 md 中运行结果如图 1-xx 所示。



平板 sm 中运行结果如图 1-xx 所示。



手机 xs 中运行结果如图 1-xx 所示。

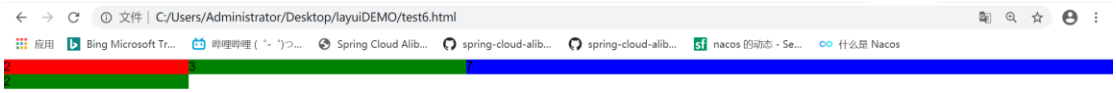
1.3.8 大于 12 等分后的列在下一行显示

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-lg2" style="background-color:
red;">2</div>
            <div class="layui-col-lg3" style="background-color:
green;">3</div>
```

```
            <div class="layui-col-lg7" style="background-color:
blue;">7</div>
            <div class="layui-col-lg2" style="background-color:
green;">2</div>
        </div>
        <span id="screenWidth"></span>
        <script>
            $(window).resize(function() {
                $("#screenWidth").text($(window).width());
            });
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



1.3.9 响应式公共类

响应式公共类如图 1-xx 所示。

类名（class）	说明
layui-show-*-block	定义不同设备下的 display: block; * 可选值有：xs、sm、md、lg
layui-show-*-inline	定义不同设备下的 display: inline; * 可选值同上
layui-show-*-inline-block	定义不同设备下的 display: inline-block; * 可选值同上
layui-hide-*	定义不同设备下的隐藏类，即： display: none; * 可选值同上

图 1-xx

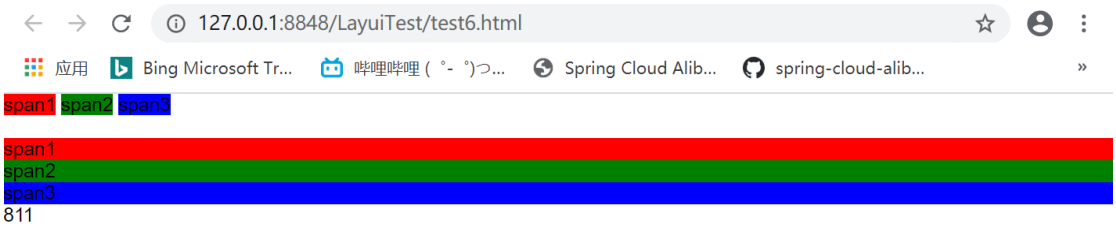
1.3.9.1 测试 layui-show-*-block

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
```

```
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-sm*">
                <span style="background-color: red;">span1</span>
                <span style="background-color: green;">span2</span>
                <span style="background-color: blue;">span3</span>
            </div>
            <br />
            <div class="layui-col-sm*" style="background-color: red;">
                <span class="layui-show-sm-block" style="background-color:
red;">span1</span>
                <span class="layui-show-sm-block" style="background-color:
green;">span2</span>
                <span class="layui-show-sm-block" style="background-color:
blue;">span3</span>
            </div>
        </div>
        <span id="screenWidth"></span>
        <script>
            $(window).resize(function() {
                $("#screenWidth").text($(window).width());
            });
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



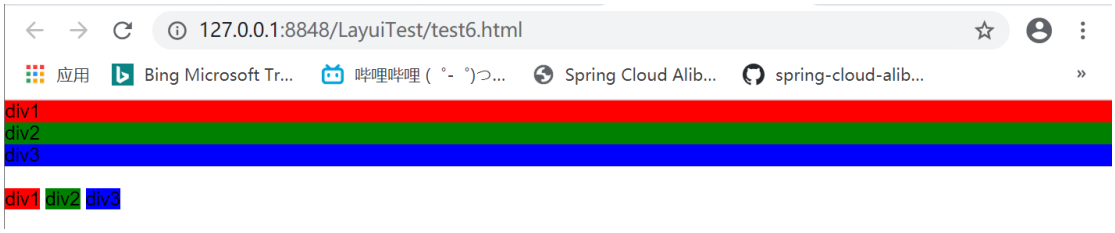
1.3.9.2 测试 layui-show-*-inline

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
```

```
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-sm*">
                <div style="background-color: red;">div1</div>
                <div style="background-color: green;">div2</div>
                <div style="background-color: blue;">div3</div>
            </div>
            <br />
            <div class="layui-col-sm*">
                <div class="layui-show-sm-inline" style="height:
100px;background-color: red;">div1</div>
                <div class="layui-show-sm-inline" style="height:
100px;background-color: green;">div2</div>
                <div class="layui-show-sm-inline" style="height:
100px;background-color: blue;">div3</div>
            </div>
        </div>
        <span id="screenWidth"></span>
        <script>
            $(window).resize(function() {
                $("#screenWidth").text($(window).width());
            });
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



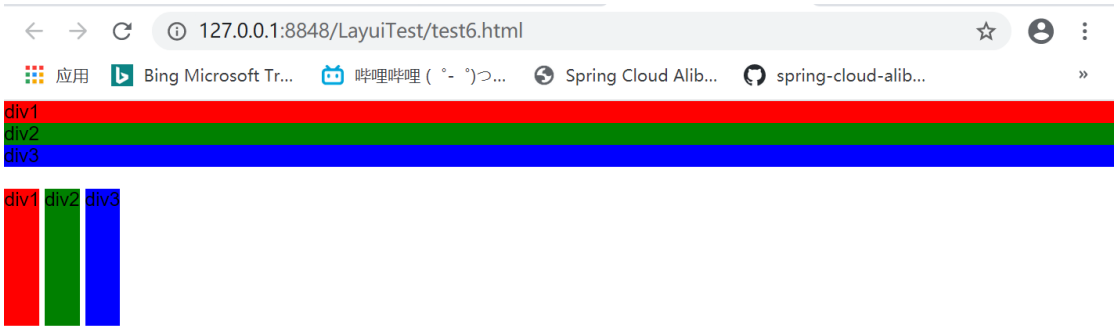
但对 div 设置的 height: 100px;属性并没有生效，需要使用 layui-show-*-inline-block 样式来使高度生效。

1.3.9.3 测试 layui-show-*-inline-block

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-sm*">
                <div style="background-color: red;">div1</div>
                <div style="background-color: green;">div2</div>
                <div style="background-color: blue;">div3</div>
            </div>
            <br />
            <div class="layui-col-sm*">
                <div class="layui-show-sm-inline-block" style="height:
100px;background-color: red;">div1</div>
                <div class="layui-show-sm-inline-block" style="height:
100px;background-color: green;">div2</div>
                <div class="layui-show-sm-inline-block" style="height:
100px;background-color: blue;">div3</div>
            </div>
        </div>
        <span id="screenWidth"></span>
        <script>
            $(window).resize(function() {
                $("#screenWidth").text($(window).width());
            });
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



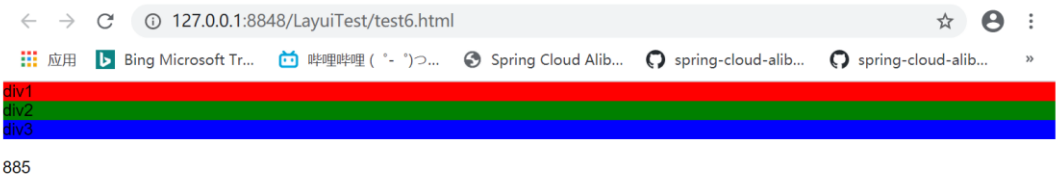
1.3.9.4 测试 layui-hide-*

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-sm*">
                <div style="background-color: red;">div1</div>
                <div style="background-color: green;">div2</div>
                <div style="background-color: blue;">div3</div>
            </div>
            <br />
            <div class="layui-col-sm*">
                <div class="layui-hide-sm" style="height:
100px;background-color: red;">div1</div>
                <div class="layui-hide-sm" style="height:
100px;background-color: green;">div2</div>
                <div class="layui-hide-sm" style="height:
100px;background-color: blue;">div3</div>
            </div>
        </div>
        <span id="screenWidth"></span>
        <script>
            $(window).resize(function() {
                $("#screenWidth").text($(window).width());
            });
```

```
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



1.3.10 设置列间距

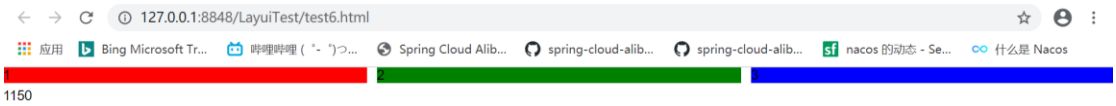
设置列间距的样式如图 1-xx 所示。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row layui-col-space10">
            <div class="layui-col-md4">
                <div style="background-color: red;">1</div>
            </div>
            <div class="layui-col-md4">
                <div style="background-color: green;">2</div>
            </div>
            <div class="layui-col-md4">
                <div style="background-color: blue;">3</div>
            </div>
        </div>
        <span id="screenWidth"></span>
        <script>
            $(window).resize(function() {
                $("#screenWidth").text($(window).width());
            });
        </script>
    </body>
```

```
</html>
```

运行结果如图 1-xx 所示。



1.3.11 设置列偏移

设置列偏移。

1.3.11.1 测试 1

对列追加 layui-col-md-offset*样式类可以让列向右偏移。其中*号代表的是偏移占据的列数，可选值为 1-12。例如 layui-col-md-offset3，代表在“中型桌面屏幕”下让该列向右偏移 3 个列宽度。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-md4" style="background-color: red;">
                4/12
            </div>
            <div class="layui-col-md4 layui-col-md-offset4"
style="background-color: green;">
                偏移4列，从而在最右
            </div>
        </div>
        <div class="layui-row">
            <div class="layui-col-md1" style="background-color: red;">
                1
            </div>
            <div class="layui-col-md1" style="background-color: green;">
                2
            </div>
```

```
            <div class="layui-col-md1" style="background-color: blue;">
                3
            </div>
            <div class="layui-col-md1" style="background-color: red;">
                4
            </div>
            <div class="layui-col-md1" style="background-color: green;">
                5
            </div>
            <div class="layui-col-md1" style="background-color: blue;">
                6
            </div>
            <div class="layui-col-md1" style="background-color: red;">
                7
            </div>
            <div class="layui-col-md1" style="background-color: green;">
                8
            </div>
            <div class="layui-col-md1" style="background-color: blue;">
                9
            </div>
            <div class="layui-col-md1" style="background-color: red;">
                10
            </div>
            <div class="layui-col-md1" style="background-color: green;">
                11
            </div>
            <div class="layui-col-md1" style="background-color: blue;">
                12
            </div>
        </div>
        <span id="screenWidth"></span>
        <script>
            $(window).resize(function() {
                $("#screenWidth").text($(window).width());
            });
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。

请注意，列偏移可针对不同屏幕的标准进行设置，比如上述的例子，只会在桌面屏幕下有效，当低于桌面屏幕规定的临界值，就会堆叠排列。

1.3.11.2 测试 2

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-md4">
                <div style="background-color: red;">4/12</div>
            </div>
            <div class="layui-col-md4 layui-col-md-offset4">
                <div style="background-color: green;">偏移4列</div>
            </div>
            <div class="layui-col-md1 layui-col-md-offset5">
                <div style="background-color: blue;">偏移5列</div>
            </div>
            <div class="layui-col-md1">
                <div style="background-color: red;">不偏移</div>
            </div>
        </div>
        <div class="layui-row">
            <div class="layui-col-md3 layui-col-md-offset3">
                <div style="background-color: green;">偏移3列</div>
            </div>
            <div class="layui-col-md3 layui-col-md-offset1">
                <div style="background-color: blue;">偏移1列</div>
            </div>
        </div>
        <br />
        <div class="layui-row">
            <div class="layui-col-md1" style="background-color: red;">
                1
            </div>
```

```
            <div class="layui-col-md1" style="background-color: green;">
                2
            </div>
            <div class="layui-col-md1" style="background-color: blue;">
                3
            </div>
            <div class="layui-col-md1" style="background-color: red;">
                4
            </div>
            <div class="layui-col-md1" style="background-color: green;">
                5
            </div>
            <div class="layui-col-md1" style="background-color: blue;">
                6
            </div>
            <div class="layui-col-md1" style="background-color: red;">
                7
            </div>
            <div class="layui-col-md1" style="background-color: green;">
                8
            </div>
            <div class="layui-col-md1" style="background-color: blue;">
                9
            </div>
            <div class="layui-col-md1" style="background-color: red;">
                10
            </div>
            <div class="layui-col-md1" style="background-color: green;">
                11
            </div>
            <div class="layui-col-md1" style="background-color: blue;">
                12
            </div>
        </div>
        <span id="screenWidth"></span>
        <script>
            $(window).resize(function() {
                $("#screenWidth").text($(window).width());
            });
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。

1.3.12 布局容器

Layui 提供两种布局容器：

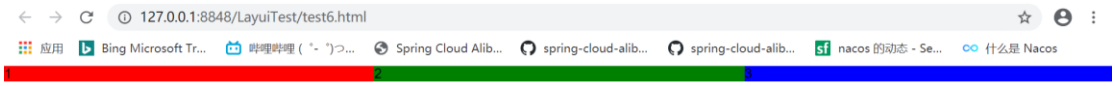
（1）class="layui-container"：将栅格放入一个带有 class="layui-container"类样式的容器中，可以在小屏幕以上的设备中固定宽度。

（2）class="layui-fluid"：将栅格或其它元素放入一个带有 class="layui-fluid"类样式的容器中，那么宽度将不会固定，而是 100%适应。

无布局容器的测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-md4">
                <div style="background-color: red;">1</div>
            </div>
            <div class="layui-col-md4">
                <div style="background-color: green;">2</div>
            </div>
            <div class="layui-col-md4">
                <div style="background-color: blue;">3</div>
            </div>
        </div>
        <span id="screenWidth"></span>
        <script>
            $(window).resize(function() {
                $("#screenWidth").text($(window).width());
            });
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。

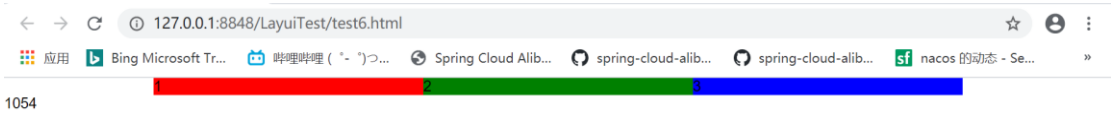


红色 div 在左上角显示，并且和浏览器边框没有间距。

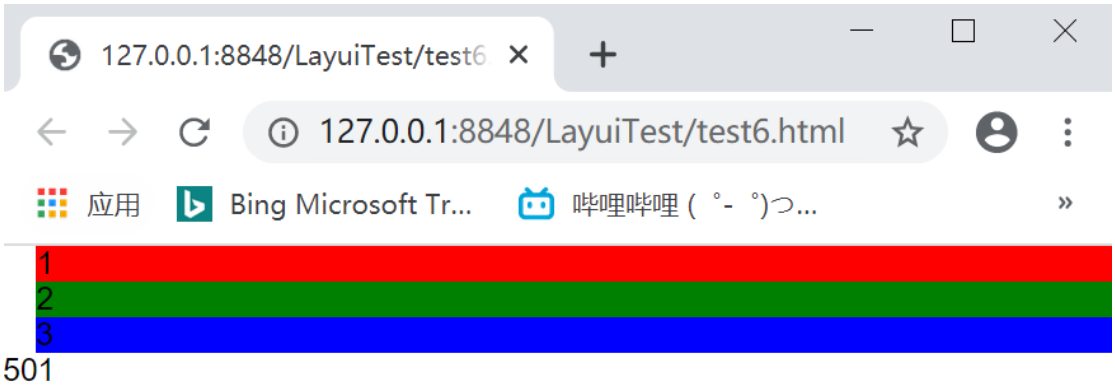
使用 class="layui-container"布局容器定义固定宽度的测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-container" style="width: 800px;">
            <div class="layui-row">
                <div class="layui-col-md4">
                    <div style="background-color: red;">1</div>
                </div>
                <div class="layui-col-md4">
                    <div style="background-color: green;">2</div>
                </div>
                <div class="layui-col-md4">
                    <div style="background-color: blue;">3</div>
                </div>
            </div>
        </div>
        <span id="screenWidth"></span>
        <script>
            $(window).resize(function() {
                $("#screenWidth").text($(window).width());
            });
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



宽度固定 800px，并且水平居中显示，并且红色 div 有左外边距，如图 1-xx 所示。

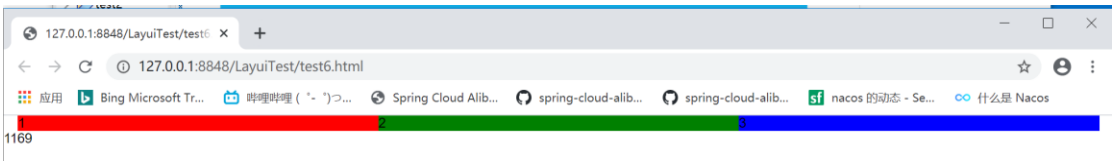


使用 class="layui-fluid"布局容器定义不是固定宽度的测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-fluid">
            <div class="layui-row">
                <div class="layui-col-md4">
                    <div style="background-color: red;">1</div>
                </div>
                <div class="layui-col-md4">
                    <div style="background-color: green;">2</div>
                </div>
                <div class="layui-col-md4">
                    <div style="background-color: blue;">3</div>
                </div>
            </div>
        </div>
        <span id="screenWidth"></span>
        <script>
            $(window).resize(function() {
                $("#screenWidth").text($(window).width());
```

```
            });
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



容器的宽度随着浏览器的宽度而改变，并且有左右外边距。

1.3.13 栅格嵌套

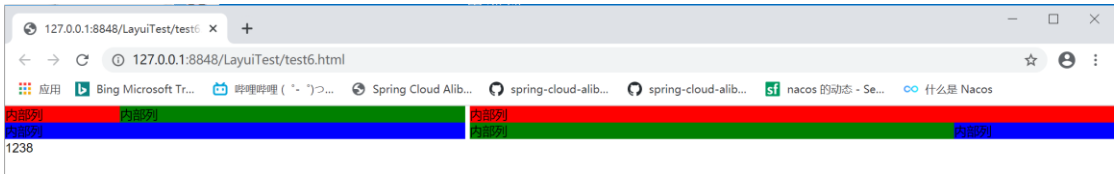
理论上可以对栅格进行无限层次的嵌套，这更加增强了栅格的表现能力。而嵌套的使用非常简单，在列元素（layui-col-md*）中插入一个行元素（layui-row）即可完成嵌套。

测试代码如下：

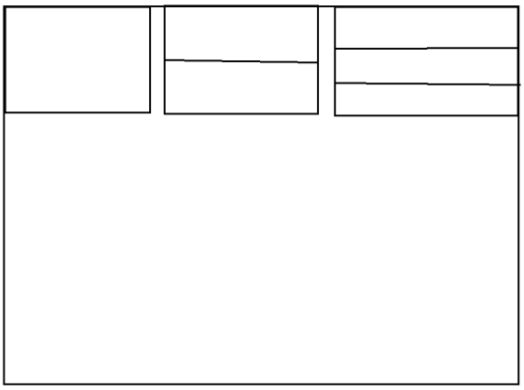
```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row layui-col-space5">
            <div class="layui-col-md5">
                <div class="layui-row">
                    <div class="layui-col-md3" style="background-color:
red;">
                        内部列
                    </div>
                    <div class="layui-col-md9" style="background-color:
green;">
                        内部列
                    </div>
                    <div class="layui-col-md12" style="background-color:
blue;">
                        内部列
                    </div>
                </div>
```

```
            </div>
            <div class="layui-col-md7">
                <div class="layui-row">
                    <div class="layui-col-md12" style="background-color:
red;">
                        内部列
                    </div>
                    <div class="layui-col-md9" style="background-color:
green;">
                        内部列
                    </div>
                    <div class="layui-col-md3" style="background-color:
blue;">
                        内部列
                    </div>
                </div>
            </div>
        </div>
        <span id="screenWidth"></span>
        <script>
            $(window).resize(function() {
                $("#screenWidth").text($(window).width());
            });
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



练习设计如图 1-xx 所示的布局。



列间距为 5 像素。

参考答案如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row layui-col-space5">
            <div class="layui-col-md4">
                <div class="layui-row">
                    <div class="layui-col-md12" style="background-color:
red;">
                        内容
                    </div>
                </div>
            </div>
            <div class="layui-col-md4">
                <div class="layui-row">
                    <div class="layui-col-md12" style="background-color:
red;">
                        内容
                    </div>
                </div>
                <div class="layui-row">
                    <div class="layui-col-md12" style="background-color:
red;">
                        内容
                    </div>
```

```
                </div>
            </div>
            <div class="layui-col-md4">
                <div class="layui-row">
                    <div class="layui-col-md12" style="background-color:
red;">
                        内容
                    </div>
                </div>
                <div class="layui-row">
                    <div class="layui-col-md12" style="background-color:
red;">
                        内容
                    </div>
                </div>
                <div class="layui-row">
                    <div class="layui-col-md12" style="background-color:
red;">
                        内容
                    </div>
                </div>
            </div>
        </div>
        <span id="screenWidth"></span>
        <script>
            $(window).resize(function() {
                $("#screenWidth").text($(window).width());
            });
        </script>
    </body>
</html>
```

1.3.14 实现 admin 后台管理布局

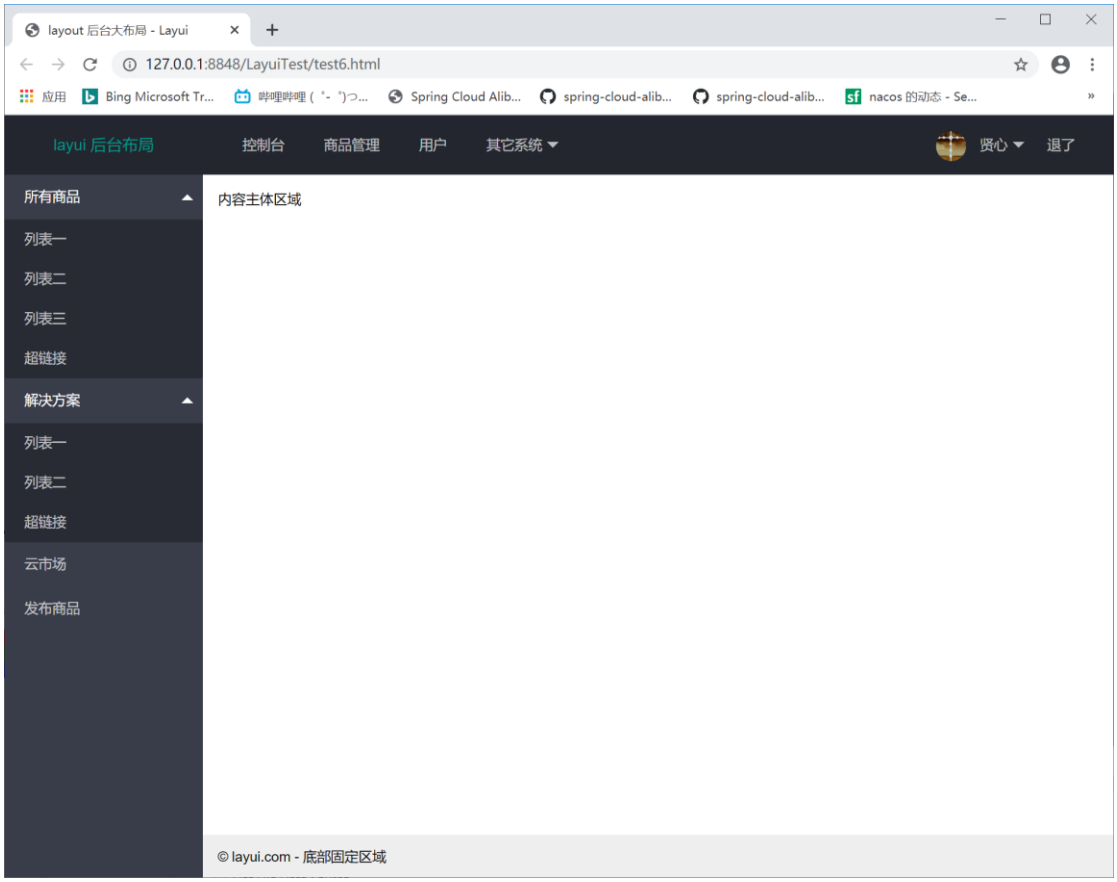
测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
    </head>
```

```
    <body class="layui-layout-body">
        <div class="layui-layout layui-layout-admin">
            <div class="layui-header">
                <div class="layui-logo">layui 后台布局</div>
                <!-- 头部区域（可配合layui已有的水平导航） -->
                <ul class="layui-nav layui-layout-left">
                    <li class="layui-nav-item"><a href="">控制台</a></li>
                    <li class="layui-nav-item"><a href="">商品管理</a></li>
                    <li class="layui-nav-item"><a href="">用户</a></li>
                    <li class="layui-nav-item">
                        <a href="javascript:;">其它系统</a>
                        <dl class="layui-nav-child">
                            <dd><a href="">邮件管理</a></dd>
                            <dd><a href="">消息管理</a></dd>
                            <dd><a href="">授权管理</a></dd>
                        </dl>
                    </li>
                </ul>
                <ul class="layui-nav layui-layout-right">
                    <li class="layui-nav-item">
                        <a href="javascript:;">
                            <img src="http://t.cn/RCzsdCq"
class="layui-nav-img">
                            贤心
                        </a>
                        <dl class="layui-nav-child">
                            <dd><a href="">基本资料</a></dd>
                            <dd><a href="">安全设置</a></dd>
                        </dl>
                    </li>
                    <li class="layui-nav-item"><a href="">退了</a></li>
                </ul>
            </div>

            <div class="layui-side layui-bg-black">
                <div class="layui-side-scroll">
                    <!-- 左侧导航区域（可配合layui已有的垂直导航） -->
                    <ul class="layui-nav layui-nav-tree" lay-filter="test">
                        <li class="layui-nav-item layui-nav-itemed">
                            <a class="" href="javascript:;">所有商品</a>
                            <dl class="layui-nav-child">
                                <dd><a href="javascript:;">列表一</a></dd>
                                <dd><a href="javascript:;">列表二</a></dd>
                                <dd><a href="javascript:;">列表三</a></dd>
```

```
                        <dd><a href="">超链接</a></dd>
                    </dl>
                </li>
                <li class="layui-nav-item">
                    <a href="javascript:;">解决方案</a>
                    <dl class="layui-nav-child">
                        <dd><a href="javascript:;">列表一</a></dd>
                        <dd><a href="javascript:;">列表二</a></dd>
                        <dd><a href="">超链接</a></dd>
                    </dl>
                </li>
                <li class="layui-nav-item"><a href="">云市场
</a></li>
                <li class="layui-nav-item"><a href="">发布商品
</a></li>
            </ul>
        </div>
    </div>

    <div class="layui-body">
        <!-- 内容主体区域 -->
        <div style="padding: 15px;">内容主体区域</div>
    </div>

    <div class="layui-footer">
        <!-- 底部固定区域 -->
        © layui.com - 底部固定区域
    </div>
</div>
<script src="layui/layui.all.js"></script>
<script>
    var element = layui.element;
</script>
</body>
</html>
```

运行结果如图 1-xx 所示。



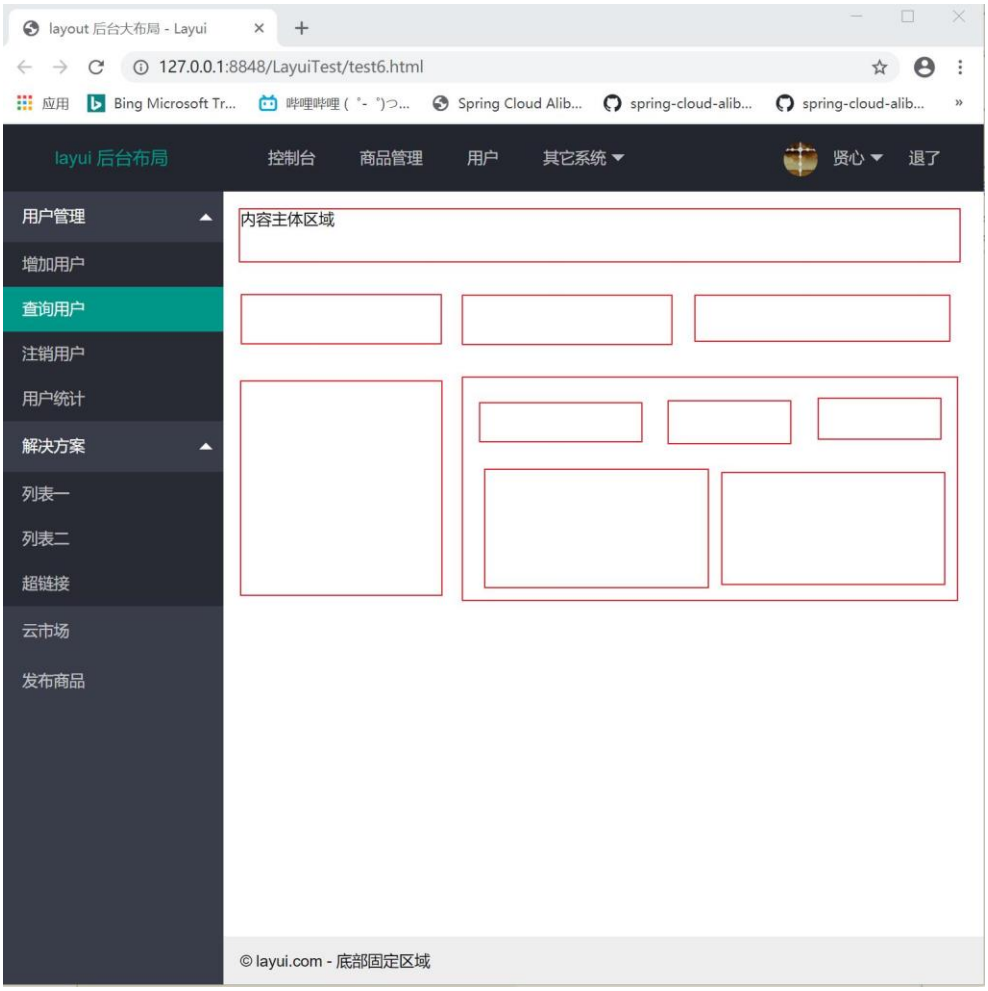
1.3.15 实现 admin 后台管理布局作业与答案

（1）再添加 2 组如图 1-xx 所示的两级分类。

（2）将“云市场”变成带下拉子类的效果。

图 1-xx

（3）设计出如图 1-xx 所示的 body 布局。



参考答案如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
    </head>
    <body class="layui-layout-body">
        <div class="layui-layout layui-layout-admin">
            <div class="layui-header">
                <div class="layui-logo">layui 后台布局</div>
                <!-- 头部区域（可配合layui已有的水平导航） -->
                <ul class="layui-nav layui-layout-left">
```

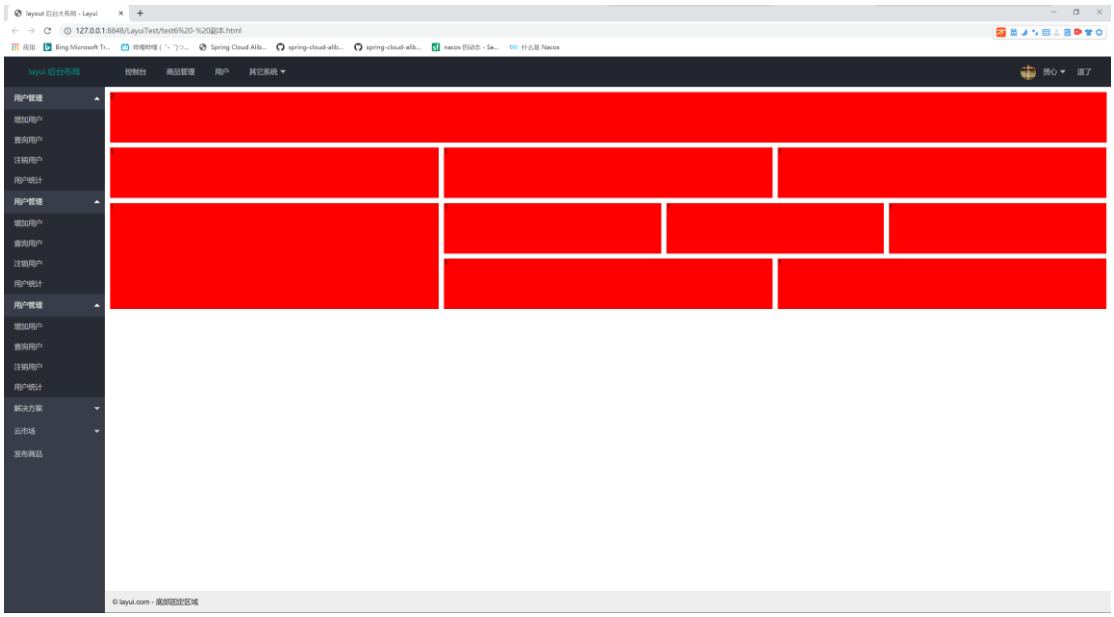
```
                <li class="layui-nav-item"><a href="">控制台</a></li>
                <li class="layui-nav-item"><a href="">商品管理</a></li>
                <li class="layui-nav-item"><a href="">用户</a></li>
                <li class="layui-nav-item">
                    <a href="javascript:;">其它系统</a>
                    <dl class="layui-nav-child">
                        <dd><a href="">邮件管理</a></dd>
                        <dd><a href="">消息管理</a></dd>
                        <dd><a href="">授权管理</a></dd>
                    </dl>
                </li>
            </ul>
            <ul class="layui-nav layui-layout-right">
                <li class="layui-nav-item">
                    <a href="javascript:;">
                        <img src="http://t.cn/RCzsdCq"
class="layui-nav-img">
                        贤心
                    </a>
                    <dl class="layui-nav-child">
                        <dd><a href="">基本资料</a></dd>
                        <dd><a href="">安全设置</a></dd>
                    </dl>
                </li>
                <li class="layui-nav-item"><a href="">退了</a></li>
            </ul>
        </div>

        <div class="layui-side layui-bg-black">
            <div class="layui-side-scroll">
                <!-- 左侧导航区域（可配合layui已有的垂直导航） -->
                <ul class="layui-nav layui-nav-tree" lay-filter="test">
                    <li class="layui-nav-item layui-nav-itemed">
                        <a class="" href="javascript:;">用户管理</a>
                        <dl class="layui-nav-child">
                            <dd><a href="javascript:;">增加用户</a></dd>
                            <dd><a href="javascript:;">查询用户</a></dd>
                            <dd><a href="javascript:;">注销用户</a></dd>
                            <dd><a href="javascript:;">用户统计</a></dd>
                        </dl>
                    </li>
                    <li class="layui-nav-item layui-nav-itemed">
                        <a class="" href="javascript:;">用户管理</a>
                        <dl class="layui-nav-child">
```

```
                        <dd><a href="javascript:;">增加用户</a></dd>
                        <dd><a href="javascript:;">查询用户</a></dd>
                        <dd><a href="javascript:;">注销用户</a></dd>
                        <dd><a href="javascript:;">用户统计</a></dd>
                    </dl>
                </li>
                <li class="layui-nav-item layui-nav-itemed">
                    <a class="" href="javascript:;">用户管理</a>
                    <dl class="layui-nav-child">
                        <dd><a href="javascript:;">增加用户</a></dd>
                        <dd><a href="javascript:;">查询用户</a></dd>
                        <dd><a href="javascript:;">注销用户</a></dd>
                        <dd><a href="javascript:;">用户统计</a></dd>
                    </dl>
                </li>
                <li class="layui-nav-item">
                    <a href="javascript:;">解决方案</a>
                    <dl class="layui-nav-child">
                        <dd><a href="javascript:;">列表一</a></dd>
                        <dd><a href="javascript:;">列表二</a></dd>
                        <dd><a href="">超链接</a></dd>
                    </dl>
                </li>
                <li class="layui-nav-item">
                    <a href="javascript:;">云市场</a>
                    <dl class="layui-nav-child">
                        <dd><a href="javascript:;">增加用户</a></dd>
                        <dd><a href="javascript:;">查询用户</a></dd>
                        <dd><a href="javascript:;">注销用户</a></dd>
                        <dd><a href="javascript:;">用户统计</a></dd>
                    </dl>
                </li>
                <li class="layui-nav-item"><a href="">发布商品
</a></li>
            </ul>
        </div>
    </div>

    <div class="layui-body">
        <div style="padding: 10px;">
            <div class="layui-row layui-col-space10">
                <div class="layui-col-md12">
                    <div style="background-color: red; height:
100px;">2</div>
```

```
                    </div>
                </div>
                <div class="layui-row layui-col-space10">
                    <div class="layui-col-md4">
                        <div style="background-color: red; height:
100px;">1</div>
                    </div>
                    <div class="layui-col-md4">
                        <div style="background-color: red; height:
100px;">1</div>
                    </div>
                    <div class="layui-col-md4">
                        <div style="background-color: red; height:
100px;">1</div>
                    </div>
                </div>
                <div class="layui-row layui-col-space10">
                    <div class="layui-col-md4">
                        <div style="background-color: red; height:
210px">1</div>
                    </div>
                    <div class="layui-col-md8">
                        <div class="layui-row layui-col-space10">
                            <div class="layui-col-md4">
                                <div style="background-color: red;
height: 100px;">1</div>
                            </div>
                            <div class="layui-col-md4">
                                <div style="background-color: red;
height: 100px;">1</div>
                            </div>
                            <div class="layui-col-md4">
                                <div style="background-color: red;
height: 100px;">1</div>
                            </div>
                        </div>
                        <div class="layui-row layui-col-space10">
                            <div class="layui-col-md6">
                                <div style="background-color: red;
height: 100px;">1</div>
                            </div>
                            <div class="layui-col-md6">
                                <div style="background-color: red;
height: 100px;">1</div>
```

```
                        </div>
                    </div>
                </div>
            </div>
        </div>

        <div class="layui-footer">
            <!-- 底部固定区域 -->
            © layui.com - 底部固定区域
        </div>
    </div>
    <script src="layui/layui.all.js"></script>
    <script>
        var element = layui.element;
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.4 颜色

视觉疲劳的形成往往是由于颜色过于丰富或过于单一形成的麻木感，而 Layui 提供的颜色不仅清新而且不乏深沉，互相中和，不过分刺激视觉神经，形成越久越耐看的微妙影像，使 Web 平台的用户界面看上去更为融洽。

1.4.1 常用主色

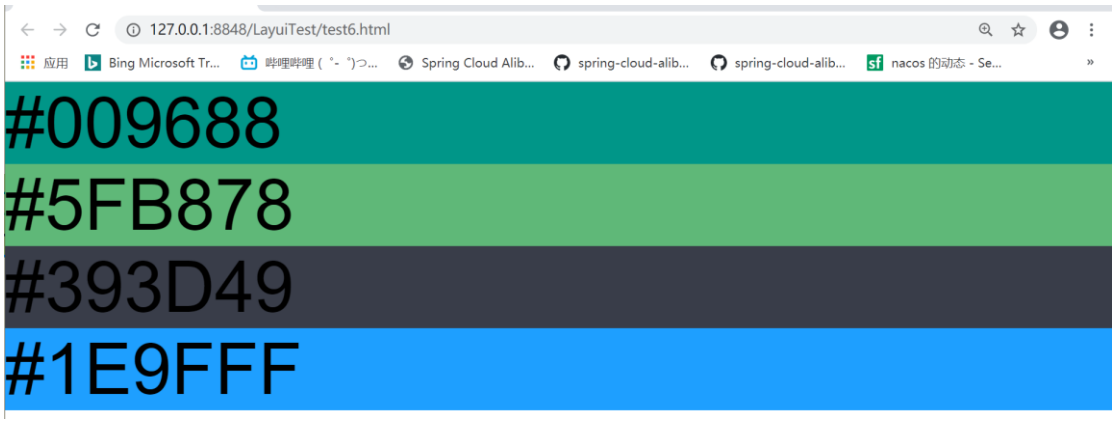
常用主色如图 1-xx 所示。

Layui 主要是以象征包容的墨绿色作为主色调，由于它给人以深沉感，所以通常会以浅黑色作为其陪衬，又会以蓝色这种比较鲜艳的色调来弥补它的色觉疲劳，整体让人清新自然，愈发耐看。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-md12" style="background-color:
#009688;">#009688</div>
            <div class="layui-col-md12" style="background-color:
#5FB878;">#5FB878</div>
            <div class="layui-col-md12" style="background-color:
#393D49;">#393D49</div>
            <div class="layui-col-md12" style="background-color:
#1E9FFF;">#1E9FFF</div>
        </div>
    </body>
</html>
```

运行结果如图 1-xx 所示。



1.4.2 场景色

场景色如图 1-xx 所示。

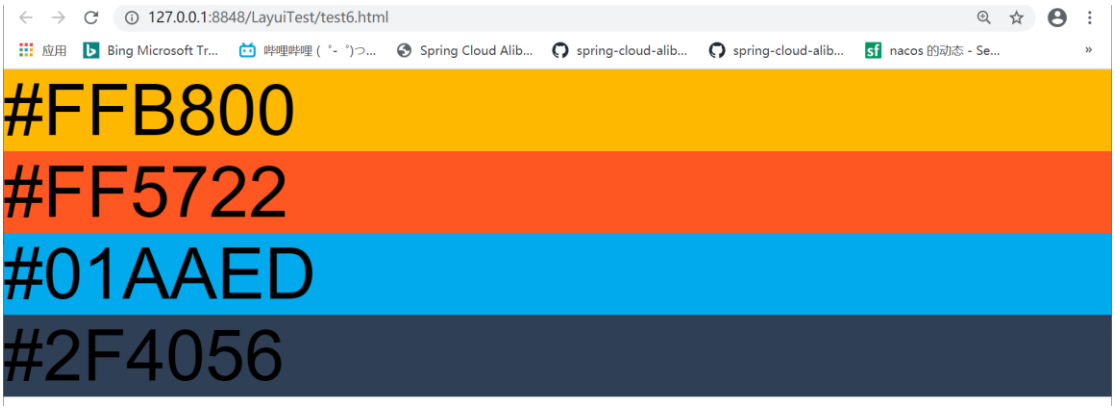
事实上，Layui 并非不敢去尝试一些亮丽的颜色，但许多情况下它可能并不是那么合适，所以把这些颜色归为“场景色”，即按照实际场景来呈现对应的颜色，比如想给人以警觉感，可以尝试用上面的红色。它们一般会出现在 Layui 的按钮、提示和修饰性元素以及一些侧边元素上。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-md12" style="background-color:
#FFB800;">#FFB800</div>
            <div class="layui-col-md12" style="background-color:
#FF5722;">#FF5722</div>
            <div class="layui-col-md12" style="background-color:
#01AAED;">#01AAED</div>
            <div class="layui-col-md12" style="background-color:
#2F4056;">#2F4056</div>
        </div>
```

```
    </body>
</html>
```

运行结果如下：



1.4.3 极简中性色

极简中性色如图 1-xx 所示。

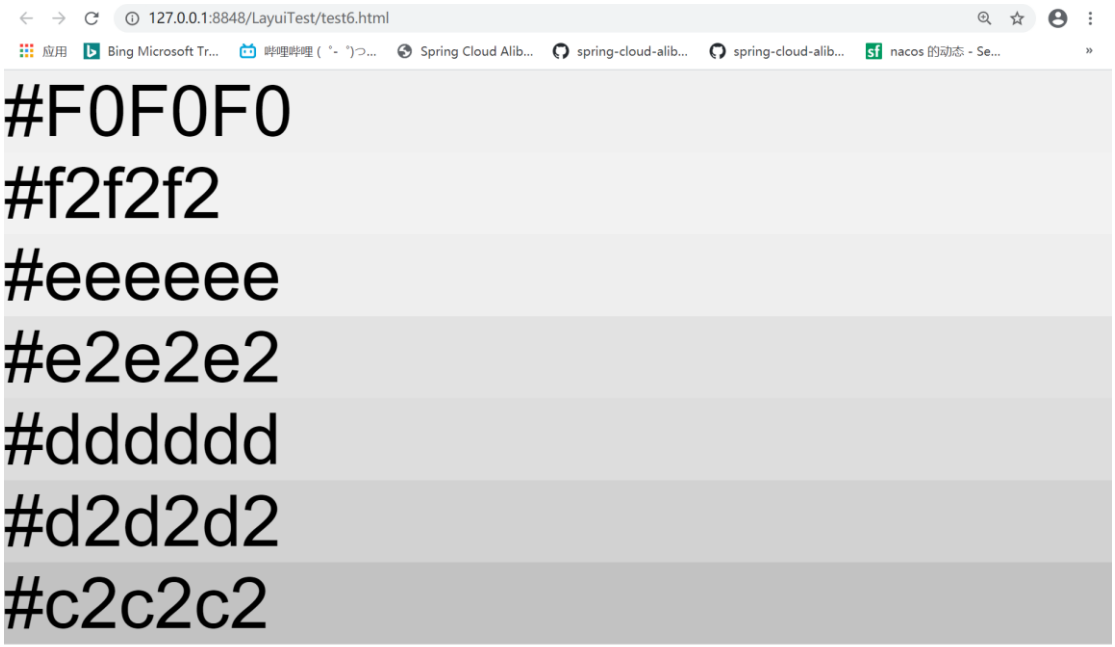
它们一般用于背景、边框等。Layui 认为灰色系代表极简，因为这是一种神奇的颜色，几乎可以与任何元素搭配，不易形成视觉疲劳，且永远不会过时。低调而优雅！

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-md12" style="background-color:
#F0F0F0;">#F0F0F0</div>
            <div class="layui-col-md12" style="background-color:
#f2f2f2;">#f2f2f2</div>
            <div class="layui-col-md12" style="background-color:
#eeeeee;">#eeeeee</div>
            <div class="layui-col-md12" style="background-color:
#e2e2e2;">#e2e2e2</div>
```

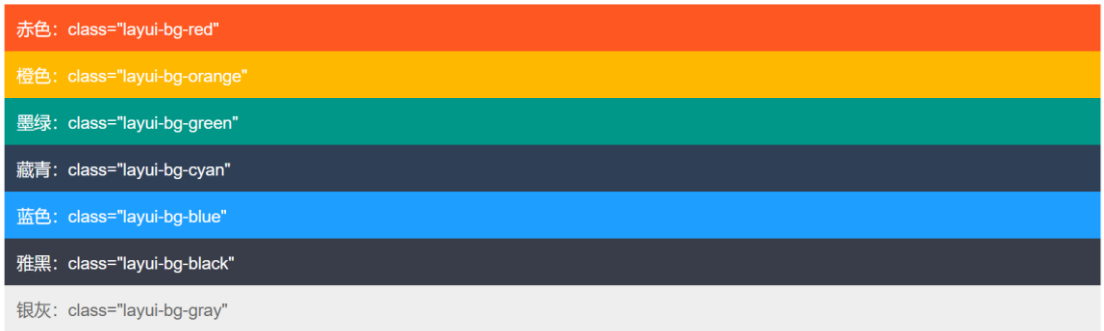
```
            <div class="layui-col-md12" style="background-color:
#dddddd;">#dddddd</div>
            <div class="layui-col-md12" style="background-color:
#d2d2d2;">#d2d2d2</div>
            <div class="layui-col-md12" style="background-color:
#c2c2c2;">#c2c2c2</div>
        </div>
    </body>
</html>
```

运行结果如图 1-xx 所示。



1.4.4 内置的背景色 CSS 样式类

内置的背景色 CSS 样式类如图 1-xx 所示。

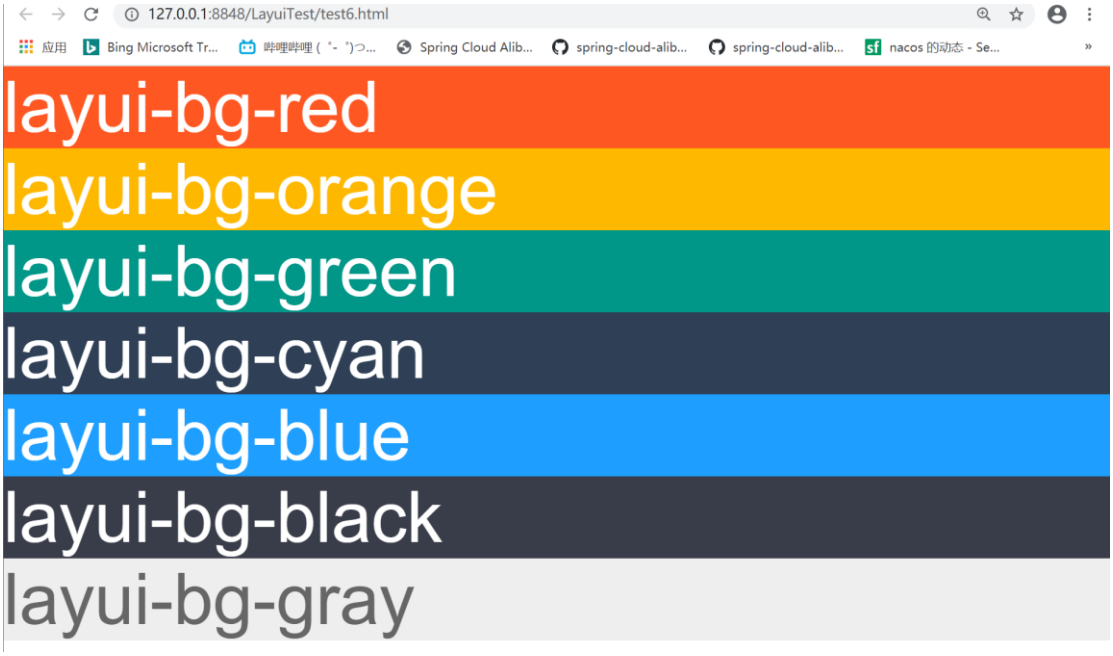


Layui 内置了七种背景色：
赤色：class="layui-bg-red"

橙色：class="layui-bg-orange"
墨绿：class="layui-bg-green"
藏青：class="layui-bg-cyan"
蓝色：class="layui-bg-blue"
雅黑：class="layui-bg-black"
银灰：class="layui-bg-gray"
以便用于各种元素，例如徽章、分割线、导航等等。
测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-md12 layui-bg-red">layui-bg-red</div>
            <div class="layui-col-md12
layui-bg-orange">layui-bg-orange</div>
            <div class="layui-col-md12
layui-bg-green">layui-bg-green</div>
            <div class="layui-col-md12 layui-bg-cyan">layui-bg-cyan</div>
            <div class="layui-col-md12 layui-bg-blue">layui-bg-blue</div>
            <div class="layui-col-md12
layui-bg-black">layui-bg-black</div>
            <div class="layui-col-md12 layui-bg-gray">layui-bg-gray</div>
        </div>
    </body>
</html>
```

运行结果如图 1-xx 所示。



1.5 字体图标

Layui 的所有图标全部采用字体形式，取材于阿里巴巴矢量图标库（iconfont）。因此可以把一个 icon 看作是一个普通的文字，这意味着直接用 css 控制文字属性，如 color、font-size 就可以改变图标的颜色和大小。可以通过 font-class 来定义不同的图标。

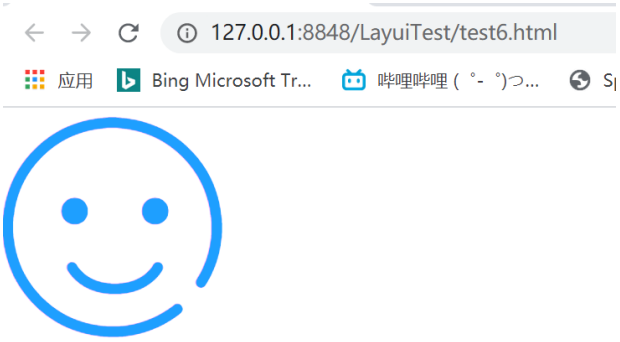
1.5.1 使用方式

通过对一个内联元素（一般推荐使用<i>标签）设置 class="layui-icon"样式来定义一个图标，然后对元素加上图标对应的 font-class 即可显示出想要的图标，测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-md12">
                <i class="layui-icon layui-icon-face-smile"
style="font-size: 30px; color: #1E9FFF;"></i>
            </div>
        </div>
    </body>
```

```
</html>
```

运行结果如图 1-xx 所示。



1.5.2 内置图标一览表（168 个）

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <style>
            ul li {
                float: left;
                margin-left: 20px;
                list-style-type: none;
            }
        </style>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-md12">
                <ul class="site-doc-icon" style="float: clear;">
                    <li>
                        <i class="layui-icon layui-icon-heart-fill"></i>
                        <div class="doc-icon-name">实心</div>
                        <div
class="doc-icon-fontclass">layui-icon-heart-fill</div>
                    </li>
                    <li>
                        <i class="layui-icon layui-icon-heart"></i>
                        <div class="doc-icon-name">空心</div>
```

```
                    <div
class="doc-icon-fontclass">layui-icon-heart</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-light"></i>
                    <div class="doc-icon-name">亮度/晴</div>
                    <div
class="doc-icon-fontclass">layui-icon-light</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-time"></i>
                    <div class="doc-icon-name">时间/历史</div>
                    <div
class="doc-icon-fontclass">layui-icon-time</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-bluetooth"></i>
                    <div class="doc-icon-name">蓝牙</div>
                    <div
class="doc-icon-fontclass">layui-icon-bluetooth</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-at"></i>
                    <div class="doc-icon-name">@艾特</div>
                    <div
class="doc-icon-fontclass">layui-icon-at</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-mute"></i>
                    <div class="doc-icon-name">静音</div>
                    <div
class="doc-icon-fontclass">layui-icon-mute</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-mike"></i>
                    <div class="doc-icon-name">录音/麦克风</div>
                    <div
class="doc-icon-fontclass">layui-icon-mike</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-key"></i>
                    <div class="doc-icon-name">密钥/钥匙</div>
                    <div
class="doc-icon-fontclass">layui-icon-key</div>
```

```
                </li>
                <li>
                    <i class="layui-icon layui-icon-gift"></i>
                    <div class="doc-icon-name">礼物/活动</div>
                    <div
class="doc-icon-fontclass">layui-icon-gift</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-email"></i>
                    <div class="doc-icon-name">邮箱</div>
                    <div
class="doc-icon-fontclass">layui-icon-email</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-rss"></i>
                    <div class="doc-icon-name">RSS</div>
                    <div
class="doc-icon-fontclass">layui-icon-rss</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-wifi"></i>
                    <div class="doc-icon-name">WiFi</div>
                    <div
class="doc-icon-fontclass">layui-icon-wifi</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-logout"></i>
                    <div class="doc-icon-name">退出/注销</div>
                    <div
class="doc-icon-fontclass">layui-icon-logout</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-android"></i>
                    <div class="doc-icon-name">Android 安卓</div>
                    <div
class="doc-icon-fontclass">layui-icon-android</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-ios"></i>
                    <div class="doc-icon-name">Apple IOS 苹果</div>
                    <div
class="doc-icon-fontclass">layui-icon-ios</div>
                </li>
                <li>
```

```
                    <i class="layui-icon layui-icon-windows"></i>
                    <div class="doc-icon-name">Windows</div>
                    <div
class="doc-icon-fontclass">layui-icon-windows</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-transfer"></i>
                    <div class="doc-icon-name">穿梭框</div>
                    <div
class="doc-icon-fontclass">layui-icon-transfer</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-service"></i>
                    <div class="doc-icon-name">客服</div>
                    <div
class="doc-icon-fontclass">layui-icon-service</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-subtraction"></i>
                    <div class="doc-icon-name">减</div>
                    <div
class="doc-icon-fontclass">layui-icon-subtraction</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-addition"></i>
                    <div class="doc-icon-name">加</div>
                    <div
class="doc-icon-fontclass">layui-icon-addition</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-slider"></i>
                    <div class="doc-icon-name">滑块</div>
                    <div
class="doc-icon-fontclass">layui-icon-slider</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-print"></i>
                    <div class="doc-icon-name">打印</div>
                    <div
class="doc-icon-fontclass">layui-icon-print</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-export"></i>
                    <div class="doc-icon-name">导出</div>
```

```
                    <div
class="doc-icon-fontclass">layui-icon-export</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-cols"></i>
                    <div class="doc-icon-name">列</div>
                    <div
class="doc-icon-fontclass">layui-icon-cols</div>
                </li>
                <li>
                    <i class="layui-icon
layui-icon-screen-restore"></i>
                    <div class="doc-icon-name">退出全屏</div>
                    <div
class="doc-icon-fontclass">layui-icon-screen-restore</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-screen-full"></i>
                    <div class="doc-icon-name">全屏</div>
                    <div
class="doc-icon-fontclass">layui-icon-screen-full</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-rate-half"></i>
                    <div class="doc-icon-name">半星</div>
                    <div
class="doc-icon-fontclass">layui-icon-rate-half</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-rate"></i>
                    <div class="doc-icon-name">星星-空心</div>
                    <div
class="doc-icon-fontclass">layui-icon-rate</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-rate-solid"></i>
                    <div class="doc-icon-name">星星-实心</div>
                    <div
class="doc-icon-fontclass">layui-icon-rate-solid</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-cellphone"></i>
                    <div class="doc-icon-name">手机</div>
                    <div
```

```
class="doc-icon-fontclass">layui-icon-cellphone</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-vercode"></i>
                    <div class="doc-icon-name">验证码</div>
                    <div
class="doc-icon-fontclass">layui-icon-vercode</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-login-wechat"></i>
                    <div class="doc-icon-name">微信</div>
                    <div
class="doc-icon-fontclass">layui-icon-login-wechat</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-login-qq"></i>
                    <div class="doc-icon-name">QQ</div>
                    <div
class="doc-icon-fontclass">layui-icon-login-qq</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-login-weibo"></i>
                    <div class="doc-icon-name">微博</div>
                    <div
class="doc-icon-fontclass">layui-icon-login-weibo</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-password"></i>
                    <div class="doc-icon-name">密码</div>
                    <div
class="doc-icon-fontclass">layui-icon-password</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-username"></i>
                    <div class="doc-icon-name">用户名</div>
                    <div
class="doc-icon-fontclass">layui-icon-username</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-refresh-3"></i>
                    <div class="doc-icon-name">刷新-粗</div>
                    <div
class="doc-icon-fontclass">layui-icon-refresh-3</div>
                </li>
```

```
                <li>
                    <i class="layui-icon layui-icon-auz"></i>
                    <div class="doc-icon-name">授权</div>
                    <div
class="doc-icon-fontclass">layui-icon-auz</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-spread-left"></i>
                    <div class="doc-icon-name">左向右伸缩菜单</div>
                    <div
class="doc-icon-fontclass">layui-icon-spread-left</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-shrink-right"></i>
                    <div class="doc-icon-name">右向左伸缩菜单</div>
                    <div
class="doc-icon-fontclass">layui-icon-shrink-right</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-snowflake"></i>
                    <div class="doc-icon-name">雪花</div>
                    <div
class="doc-icon-fontclass">layui-icon-snowflake</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-tips"></i>
                    <div class="doc-icon-name">提示说明</div>
                    <div
class="doc-icon-fontclass">layui-icon-tips</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-note"></i>
                    <div class="doc-icon-name">便签</div>
                    <div
class="doc-icon-fontclass">layui-icon-note</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-home"></i>
                    <div class="doc-icon-name">主页</div>
                    <div
class="doc-icon-fontclass">layui-icon-home</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-senior"></i>
```

```
                    <div class="doc-icon-name">高级</div>
                    <div
class="doc-icon-fontclass">layui-icon-senior</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-refresh"></i>
                    <div class="doc-icon-name">刷新</div>
                    <div
class="doc-icon-fontclass">layui-icon-refresh</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-refresh-1"></i>
                    <div class="doc-icon-name">刷新</div>
                    <div
class="doc-icon-fontclass">layui-icon-refresh-1</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-flag"></i>
                    <div class="doc-icon-name">旗帜</div>
                    <div
class="doc-icon-fontclass">layui-icon-flag</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-theme"></i>
                    <div class="doc-icon-name">主题</div>
                    <div
class="doc-icon-fontclass">layui-icon-theme</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-notice"></i>
                    <div class="doc-icon-name">消息-通知</div>
                    <div
class="doc-icon-fontclass">layui-icon-notice</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-website"></i>
                    <div class="doc-icon-name">网站</div>
                    <div
class="doc-icon-fontclass">layui-icon-website</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-console"></i>
                    <div class="doc-icon-name">控制台</div>
                    <div
```

```
class="doc-icon-fontclass">layui-icon-console</div>
                </li>
                <li>
                    <i class="layui-icon
layui-icon-face-surprised"></i>
                    <div class="doc-icon-name">表情-惊讶</div>
                    <div
class="doc-icon-fontclass">layui-icon-face-surprised</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-set"></i>
                    <div class="doc-icon-name">设置-空心</div>
                    <div
class="doc-icon-fontclass">layui-icon-set</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-template-1"></i>
                    <div class="doc-icon-name">模板</div>
                    <div
class="doc-icon-fontclass">layui-icon-template-1</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-app"></i>
                    <div class="doc-icon-name">应用</div>
                    <div
class="doc-icon-fontclass">layui-icon-app</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-template"></i>
                    <div class="doc-icon-name">模板</div>
                    <div
class="doc-icon-fontclass">layui-icon-template</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-praise"></i>
                    <div class="doc-icon-name">赞</div>
                    <div
class="doc-icon-fontclass">layui-icon-praise</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-tread"></i>
                    <div class="doc-icon-name">踩</div>
                    <div
class="doc-icon-fontclass">layui-icon-tread</div>
```

```
                </li>
                <li>
                    <i class="layui-icon layui-icon-male"></i>
                    <div class="doc-icon-name">男</div>
                    <div
class="doc-icon-fontclass">layui-icon-male</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-female"></i>
                    <div class="doc-icon-name">女</div>
                    <div
class="doc-icon-fontclass">layui-icon-female</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-camera"></i>
                    <div class="doc-icon-name">相机-空心</div>
                    <div
class="doc-icon-fontclass">layui-icon-camera</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-camera-fill"></i>
                    <div class="doc-icon-name">相机-实心</div>
                    <div
class="doc-icon-fontclass">layui-icon-camera-fill</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-more"></i>
                    <div class="doc-icon-name">菜单-水平</div>
                    <div
class="doc-icon-fontclass">layui-icon-more</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-more-vertical"></i>
                    <div class="doc-icon-name">菜单-垂直</div>
                    <div
class="doc-icon-fontclass">layui-icon-more-vertical</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-rmb"></i>
                    <div class="doc-icon-name">金额-人民币</div>
                    <div
class="doc-icon-fontclass">layui-icon-rmb</div>
                </li>
                <li>
```

```
                    <i class="layui-icon layui-icon-dollar"></i>
                    <div class="doc-icon-name">金额-美元</div>
                    <div
class="doc-icon-fontclass">layui-icon-dollar</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-diamond"></i>
                    <div class="doc-icon-name">钻石-等级</div>
                    <div
class="doc-icon-fontclass">layui-icon-diamond</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-fire"></i>
                    <div class="doc-icon-name">火</div>
                    <div
class="doc-icon-fontclass">layui-icon-fire</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-return"></i>
                    <div class="doc-icon-name">返回</div>
                    <div
class="doc-icon-fontclass">layui-icon-return</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-location"></i>
                    <div class="doc-icon-name">位置-地图</div>
                    <div
class="doc-icon-fontclass">layui-icon-location</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-read"></i>
                    <div class="doc-icon-name">办公-阅读</div>
                    <div
class="doc-icon-fontclass">layui-icon-read</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-survey"></i>
                    <div class="doc-icon-name">调查</div>
                    <div
class="doc-icon-fontclass">layui-icon-survey</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-face-smile"></i>
                    <div class="doc-icon-name">表情-微笑</div>
```

```
                    <div
class="doc-icon-fontclass">layui-icon-face-smile</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-face-cry"></i>
                    <div class="doc-icon-name">表情-哭泣</div>
                    <div
class="doc-icon-fontclass">layui-icon-face-cry</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-cart-simple"></i>
                    <div class="doc-icon-name">购物车</div>
                    <div
class="doc-icon-fontclass">layui-icon-cart-simple</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-cart"></i>
                    <div class="doc-icon-name">购物车</div>
                    <div
class="doc-icon-fontclass">layui-icon-cart</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-next"></i>
                    <div class="doc-icon-name">下一页</div>
                    <div
class="doc-icon-fontclass">layui-icon-next</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-prev"></i>
                    <div class="doc-icon-name">上一页</div>
                    <div
class="doc-icon-fontclass">layui-icon-prev</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-upload-drag"></i>
                    <div class="doc-icon-name">上传-空心-拖拽</div>
                    <div
class="doc-icon-fontclass">layui-icon-upload-drag</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-upload"></i>
                    <div class="doc-icon-name">上传-实心</div>
                    <div
class="doc-icon-fontclass">layui-icon-upload</div>
```

```
                </li>
                <li>
                    <i class="layui-icon
layui-icon-download-circle"></i>
                    <div class="doc-icon-name">下载-圆圈</div>
                    <div
class="doc-icon-fontclass">layui-icon-download-circle</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-component"></i>
                    <div class="doc-icon-name">组件</div>
                    <div
class="doc-icon-fontclass">layui-icon-component</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-file-b"></i>
                    <div class="doc-icon-name">文件-粗</div>
                    <div
class="doc-icon-fontclass">layui-icon-file-b</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-user"></i>
                    <div class="doc-icon-name">用户</div>
                    <div
class="doc-icon-fontclass">layui-icon-user</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-find-fill"></i>
                    <div class="doc-icon-name">发现-实心</div>
                    <div
class="doc-icon-fontclass">layui-icon-find-fill</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-loading layui-icon
layui-anim layui-anim-rotate layui-anim-loop"></i>
                    <div class="doc-icon-name">loading</div>
                    <div
class="doc-icon-fontclass">layui-icon-loading</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-loading-1
layui-icon layui-anim layui-anim-rotate layui-anim-loop"></i>
                    <div class="doc-icon-name">loading</div>
                    <div
```

```
class="doc-icon-fontclass">layui-icon-loading-1</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-add-1"></i>
                    <div class="doc-icon-name">添加</div>
                    <div
class="doc-icon-fontclass">layui-icon-add-1</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-play"></i>
                    <div class="doc-icon-name">播放</div>
                    <div
class="doc-icon-fontclass">layui-icon-play</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-pause"></i>
                    <div class="doc-icon-name">暂停</div>
                    <div
class="doc-icon-fontclass">layui-icon-pause</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-headset"></i>
                    <div class="doc-icon-name">音频-耳机</div>
                    <div
class="doc-icon-fontclass">layui-icon-headset</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-video"></i>
                    <div class="doc-icon-name">视频</div>
                    <div
class="doc-icon-fontclass">layui-icon-video</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-voice"></i>
                    <div class="doc-icon-name">语音-声音</div>
                    <div
class="doc-icon-fontclass">layui-icon-voice</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-speaker"></i>
                    <div class="doc-icon-name">消息-通知-喇叭</div>
                    <div
class="doc-icon-fontclass">layui-icon-speaker</div>
                </li>
```

```
                <li>
                    <i class="layui-icon layui-icon-fonts-del"></i>
                    <div class="doc-icon-name">删除线</div>
                    <div
class="doc-icon-fontclass">layui-icon-fonts-del</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-fonts-code"></i>
                    <div class="doc-icon-name">代码</div>
                    <div
class="doc-icon-fontclass">layui-icon-fonts-code</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-fonts-html"></i>
                    <div class="doc-icon-name">HTML</div>
                    <div
class="doc-icon-fontclass">layui-icon-fonts-html</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-fonts-strong"></i>
                    <div class="doc-icon-name">字体加粗</div>
                    <div
class="doc-icon-fontclass">layui-icon-fonts-strong</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-unlink"></i>
                    <div class="doc-icon-name">删除链接</div>
                    <div
class="doc-icon-fontclass">layui-icon-unlink</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-picture"></i>
                    <div class="doc-icon-name">图片</div>
                    <div
class="doc-icon-fontclass">layui-icon-picture</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-link"></i>
                    <div class="doc-icon-name">链接</div>
                    <div
class="doc-icon-fontclass">layui-icon-link</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-face-smile-b"></i>
```

```
                    <div class="doc-icon-name">表情-笑-粗</div>
                    <div
class="doc-icon-fontclass">layui-icon-face-smile-b</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-align-left"></i>
                    <div class="doc-icon-name">左对齐</div>
                    <div
class="doc-icon-fontclass">layui-icon-align-left</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-align-right"></i>
                    <div class="doc-icon-name">右对齐</div>
                    <div
class="doc-icon-fontclass">layui-icon-align-right</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-align-center"></i>
                    <div class="doc-icon-name">居中对齐</div>
                    <div
class="doc-icon-fontclass">layui-icon-align-center</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-fonts-u"></i>
                    <div class="doc-icon-name">字体-下划线</div>
                    <div
class="doc-icon-fontclass">layui-icon-fonts-u</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-fonts-i"></i>
                    <div class="doc-icon-name">字体-斜体</div>
                    <div
class="doc-icon-fontclass">layui-icon-fonts-i</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-tabs"></i>
                    <div class="doc-icon-name">Tabs 选项卡</div>
                    <div
class="doc-icon-fontclass">layui-icon-tabs</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-radio"></i>
                    <div class="doc-icon-name">单选框-选中</div>
                    <div
```

```
class="doc-icon-fontclass">layui-icon-radio</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-circle"></i>
                    <div class="doc-icon-name">单选框-候选</div>
                    <div
class="doc-icon-fontclass">layui-icon-circle</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-edit"></i>
                    <div class="doc-icon-name">编辑</div>
                    <div
class="doc-icon-fontclass">layui-icon-edit</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-share"></i>
                    <div class="doc-icon-name">分享</div>
                    <div
class="doc-icon-fontclass">layui-icon-share</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-delete"></i>
                    <div class="doc-icon-name">删除</div>
                    <div
class="doc-icon-fontclass">layui-icon-delete</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-form"></i>
                    <div class="doc-icon-name">表单</div>
                    <div
class="doc-icon-fontclass">layui-icon-form</div>
                </li>
                <li>
                    <i class="layui-icon
layui-icon-cellphone-fine"></i>
                    <div class="doc-icon-name">手机-细体</div>
                    <div
class="doc-icon-fontclass">layui-icon-cellphone-fine</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-dialogue"></i>
                    <div class="doc-icon-name">聊天 对话 沟通</div>
                    <div
class="doc-icon-fontclass">layui-icon-dialogue</div>
```

```
                </li>
                <li>
                    <i class="layui-icon layui-icon-fonts-clear"></i>
                    <div class="doc-icon-name">文字格式化</div>
                    <div
class="doc-icon-fontclass">layui-icon-fonts-clear</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-layer"></i>
                    <div class="doc-icon-name">窗口</div>
                    <div
class="doc-icon-fontclass">layui-icon-layer</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-date"></i>
                    <div class="doc-icon-name">日期</div>
                    <div
class="doc-icon-fontclass">layui-icon-date</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-water"></i>
                    <div class="doc-icon-name">水 下雨</div>
                    <div
class="doc-icon-fontclass">layui-icon-water</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-code-circle"></i>
                    <div class="doc-icon-name">代码-圆圈</div>
                    <div
class="doc-icon-fontclass">layui-icon-code-circle</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-carousel"></i>
                    <div class="doc-icon-name">轮播组图</div>
                    <div
class="doc-icon-fontclass">layui-icon-carousel</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-prev-circle"></i>
                    <div class="doc-icon-name">翻页</div>
                    <div
class="doc-icon-fontclass">layui-icon-prev-circle</div>
                </li>
                <li>
```

```
                    <i class="layui-icon layui-icon-layouts"></i>
                    <div class="doc-icon-name">布局</div>
                    <div
class="doc-icon-fontclass">layui-icon-layouts</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-util"></i>
                    <div class="doc-icon-name">工具</div>
                    <div
class="doc-icon-fontclass">layui-icon-util</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-templeate-1"></i>
                    <div class="doc-icon-name">选择模板</div>
                    <div
class="doc-icon-fontclass">layui-icon-templeate-1</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-upload-circle"></i>
                    <div class="doc-icon-name">上传-圆圈</div>
                    <div
class="doc-icon-fontclass">layui-icon-upload-circle</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-tree"></i>
                    <div class="doc-icon-name">树</div>
                    <div
class="doc-icon-fontclass">layui-icon-tree</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-table"></i>
                    <div class="doc-icon-name">表格</div>
                    <div
class="doc-icon-fontclass">layui-icon-table</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-chart"></i>
                    <div class="doc-icon-name">图表</div>
                    <div
class="doc-icon-fontclass">layui-icon-chart</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-chart-screen"></i>
                    <div class="doc-icon-name">图标 报表 屏幕</div>
```

```
                    <div
class="doc-icon-fontclass">layui-icon-chart-screen</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-engine"></i>
                    <div class="doc-icon-name">引擎</div>
                    <div
class="doc-icon-fontclass">layui-icon-engine</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-triangle-d"></i>
                    <div class="doc-icon-name">下三角</div>
                    <div
class="doc-icon-fontclass">layui-icon-triangle-d</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-triangle-r"></i>
                    <div class="doc-icon-name">右三角</div>
                    <div
class="doc-icon-fontclass">layui-icon-triangle-r</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-file"></i>
                    <div class="doc-icon-name">文件</div>
                    <div
class="doc-icon-fontclass">layui-icon-file</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-set-sm"></i>
                    <div class="doc-icon-name">设置-小型</div>
                    <div
class="doc-icon-fontclass">layui-icon-set-sm</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-reduce-circle"></i>
                    <div class="doc-icon-name">减少-圆圈</div>
                    <div
class="doc-icon-fontclass">layui-icon-reduce-circle</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-add-circle"></i>
                    <div class="doc-icon-name">添加-圆圈</div>
                    <div
class="doc-icon-fontclass">layui-icon-add-circle</div>
```

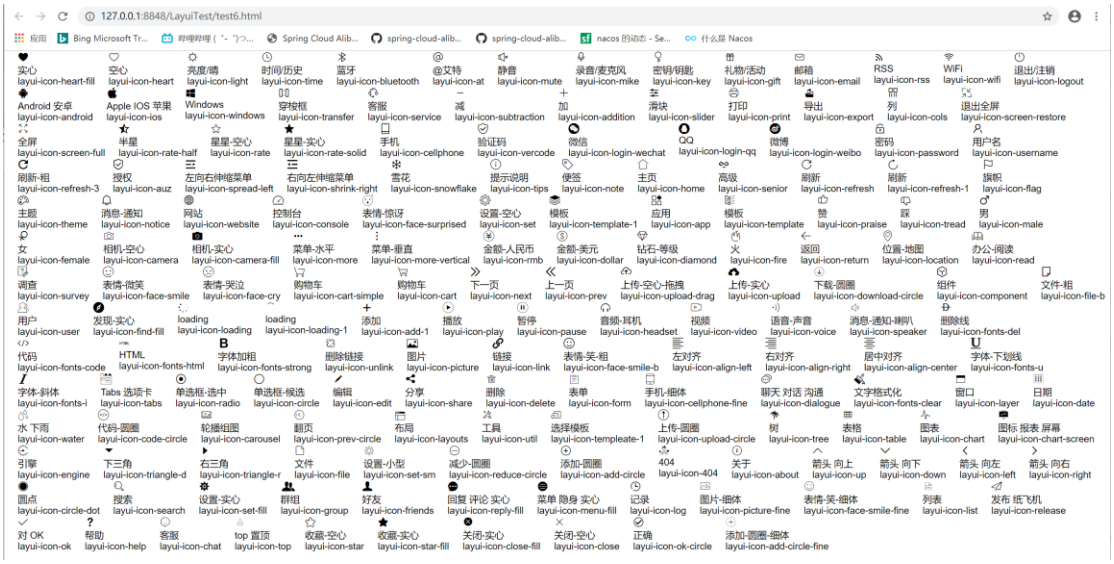
```
                </li>
                <li>
                    <i class="layui-icon layui-icon-404"></i>
                    <div class="doc-icon-name">404</div>
                    <div
class="doc-icon-fontclass">layui-icon-404</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-about"></i>
                    <div class="doc-icon-name">关于</div>
                    <div
class="doc-icon-fontclass">layui-icon-about</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-up"></i>
                    <div class="doc-icon-name">箭头 向上</div>
                    <div
class="doc-icon-fontclass">layui-icon-up</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-down"></i>
                    <div class="doc-icon-name">箭头 向下</div>
                    <div
class="doc-icon-fontclass">layui-icon-down</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-left"></i>
                    <div class="doc-icon-name">箭头 向左</div>
                    <div
class="doc-icon-fontclass">layui-icon-left</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-right"></i>
                    <div class="doc-icon-name">箭头 向右</div>
                    <div
class="doc-icon-fontclass">layui-icon-right</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-circle-dot"></i>
                    <div class="doc-icon-name">圆点</div>
                    <div
class="doc-icon-fontclass">layui-icon-circle-dot</div>
                </li>
                <li>
```

```
                    <i class="layui-icon layui-icon-search"></i>
                    <div class="doc-icon-name">搜索</div>
                    <div
class="doc-icon-fontclass">layui-icon-search</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-set-fill"></i>
                    <div class="doc-icon-name">设置-实心</div>
                    <div
class="doc-icon-fontclass">layui-icon-set-fill</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-group"></i>
                    <div class="doc-icon-name">群组</div>
                    <div
class="doc-icon-fontclass">layui-icon-group</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-friends"></i>
                    <div class="doc-icon-name">好友</div>
                    <div
class="doc-icon-fontclass">layui-icon-friends</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-reply-fill"></i>
                    <div class="doc-icon-name">回复 评论 实心</div>
                    <div
class="doc-icon-fontclass">layui-icon-reply-fill</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-menu-fill"></i>
                    <div class="doc-icon-name">菜单 隐身 实心</div>
                    <div
class="doc-icon-fontclass">layui-icon-menu-fill</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-log"></i>
                    <div class="doc-icon-name">记录</div>
                    <div
class="doc-icon-fontclass">layui-icon-log</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-picture-fine"></i>
                    <div class="doc-icon-name">图片-细体</div>
```

```
                    <div
class="doc-icon-fontclass">layui-icon-picture-fine</div>
                </li>
                <li>
                    <i class="layui-icon
layui-icon-face-smile-fine"></i>
                    <div class="doc-icon-name">表情-笑-细体</div>
                    <div
class="doc-icon-fontclass">layui-icon-face-smile-fine</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-list"></i>
                    <div class="doc-icon-name">列表</div>
                    <div
class="doc-icon-fontclass">layui-icon-list</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-release"></i>
                    <div class="doc-icon-name">发布 纸飞机</div>
                    <div
class="doc-icon-fontclass">layui-icon-release</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-ok"></i>
                    <div class="doc-icon-name">对 OK</div>
                    <div
class="doc-icon-fontclass">layui-icon-ok</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-help"></i>
                    <div class="doc-icon-name">帮助</div>
                    <div
class="doc-icon-fontclass">layui-icon-help</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-chat"></i>
                    <div class="doc-icon-name">客服</div>
                    <div
class="doc-icon-fontclass">layui-icon-chat</div>
                </li>
                <li>
                    <i class="layui-icon layui-icon-top"></i>
                    <div class="doc-icon-name">top 置顶</div>
                    <div
```

```
class="doc-icon-fontclass">layui-icon-top</div>
                    </li>
                    <li>
                        <i class="layui-icon layui-icon-star"></i>
                        <div class="doc-icon-name">收藏-空心</div>
                        <div
class="doc-icon-fontclass">layui-icon-star</div>
                    </li>
                    <li>
                        <i class="layui-icon layui-icon-star-fill"></i>
                        <div class="doc-icon-name">收藏-实心</div>
                        <div
class="doc-icon-fontclass">layui-icon-star-fill</div>
                    </li>
                    <li>
                        <i class="layui-icon layui-icon-close-fill"></i>
                        <div class="doc-icon-name">关闭-实心</div>
                        <div
class="doc-icon-fontclass">layui-icon-close-fill</div>
                    </li>
                    <li>
                        <i class="layui-icon layui-icon-close"></i>
                        <div class="doc-icon-name">关闭-空心</div>
                        <div
class="doc-icon-fontclass">layui-icon-close</div>
                    </li>
                    <li>
                        <i class="layui-icon layui-icon-ok-circle"></i>
                        <div class="doc-icon-name">正确</div>
                        <div
class="doc-icon-fontclass">layui-icon-ok-circle</div>
                    </li>
                    <li>
                        <i class="layui-icon
layui-icon-add-circle-fine"></i>
                        <div class="doc-icon-name">添加-圆圈-细体</div>
                        <div
class="doc-icon-fontclass">layui-icon-add-circle-fine</div>
                    </li>
                </ul>
            </div>
        </div>
    </body>
</html>
```

运行结果如图 1-xx 所示。



1.6 CSS3 动画类

在以实用的前提下，Layui 并没有内置过多花俏的动画。而他们同样在 Layui 的许多交互元素中发挥着重要的作用。Layui 的动画全部采用 CSS3 实现，因此不支持 ie8 和部分不支持 ie9（即在 ie8/9 中无动画效果）。

1.6.1 使用方式

动画的使用非常简单，直接对元素赋值动画特定的 class 类名即可。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-md12">
                其中 layui-anim 是必须的，后面跟着的即是不同的动画类
                <br />
                <div class="layui-anim layui-anim-up">a<br />b<br />c<br
/></div>
```

```
                循环动画，追加：layui-anim-loop
                <br />
                <div class="layui-anim layui-anim-up layui-anim-loop">1<br
/>2<br />3<br /></div>
            </div>
        </div>
    </body>
</html>
```

1.6.2 内置 CSS3 动画一览表

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-md12">
                <div id="div1" class="layui-anim layui-anim-up"
style="background-color: red;">layui-anim-up从最底部往上滑入</div>
                <br />
                <div id="div2" class="layui-anim layui-anim-upbit"
style="background-color: green;">layui-anim-upbit微微往上滑入</div>
                <br />
                <div id="div3" class="layui-anim layui-anim-scale"
style="background-color: blue;">layui-anim-scale平滑放大</div>
                <br />
                <div id="div4" class="layui-anim layui-anim-scaleSpring"
style="background-color: red;">layui-anim-scaleSpring弹簧式放大</div>
                <br />
                <div id="div5" class="layui-anim layui-anim-fadein"
style="background-color: green;">layui-anim-fadein渐现</div>
                <br />
                <div id="div6" class="layui-anim layui-anim-fadeout"
style="background-color: blue;">layui-anim-fadeout渐隐</div>
                <br />
                <div id="div7" class="layui-anim layui-anim-rotate"
```

```
style="background-color: red;">layui-anim-rotate360度旋转</div>
                <br />
                <div id="div8" class="layui-anim layui-anim-up
layui-anim-loop" style="background-color: green;">追加：layui-anim-loop循
环动画</div>
            </div>
        </div>
        <script>
            $("#div1").click(function() {
                var othis = $(this);
                $(othis).removeClass("layui-anim-up");
                setTimeout(function() {
                    $(othis).addClass("layui-anim-up");
                });
            });
            $("#div2").click(function() {
                var othis = $(this);
                $(othis).removeClass("layui-anim-upbit");
                setTimeout(function() {
                    $(othis).addClass("layui-anim-upbit");
                });
            });
            $("#div3").click(function() {
                var othis = $(this);
                $(othis).removeClass("layui-anim-scale");
                setTimeout(function() {
                    $(othis).addClass("layui-anim-scale");
                });
            });
            $("#div4").click(function() {
                var othis = $(this);
                $(othis).removeClass("layui-anim-scaleSpring");
                setTimeout(function() {
                    $(othis).addClass("layui-anim-scaleSpring");
                });
            });
            $("#div5").click(function() {
                var othis = $(this);
                $(othis).removeClass("layui-anim-fadein");
                setTimeout(function() {
                    $(othis).addClass("layui-anim-fadein");
                });
            });
            $("#div6").click(function() {
```

```
            var othis = $(this);
            $(othis).removeClass("layui-anim-fadeout");
            setTimeout(function() {
                $(othis).addClass("layui-anim-fadeout");
            });
        });
        $("#div7").click(function() {
            var othis = $(this);
            $(othis).removeClass("layui-anim-rotate");
            setTimeout(function() {
                $(othis).addClass("layui-anim-rotate");
            });
        });
        $("#div8").click(function() {
            var othis = $(this);

            if ($(othis).hasClass("layui-anim-loop")) {
                $(othis).removeClass("layui-anim-loop");
                return;
            }

            $(othis).removeClass("layui-anim-up");

            setTimeout(function() {
                $(othis).addClass("layui-anim-up");
                $(othis).addClass("layui-anim-loop");
            });
        });
    </script>
</body>
</html>
```

1.7 按钮

向任意 HTML 元素设置 class="layui-btn"来创建一个基础按钮。通过追加格式为 layui-btn-{type}的 class 来定义其它按钮风格。内置的按钮 class 可以进行任意组合，从而形成多种按钮风格。

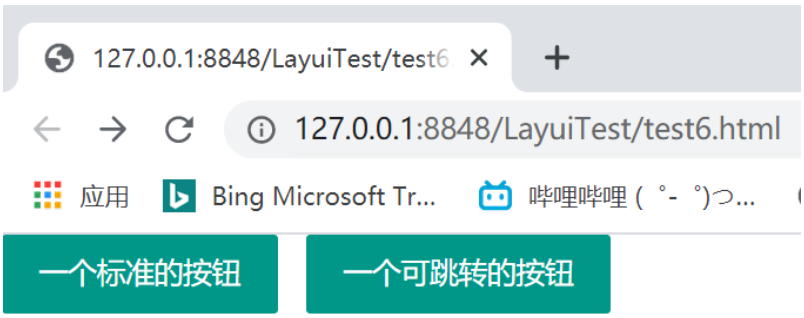
1.7.1 用法

测试代码如下：

```
<!DOCTYPE html>
<html>
```

```
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-md12">
                <button type="button" class="layui-btn">一个标准的按钮
</button>
                <a href="http://www.layui.com" class="layui-btn">一个可跳转
的按钮</a>
            </div>
        </div>
    </body>
</html>
```

运行效果如图 1-xx 所示。



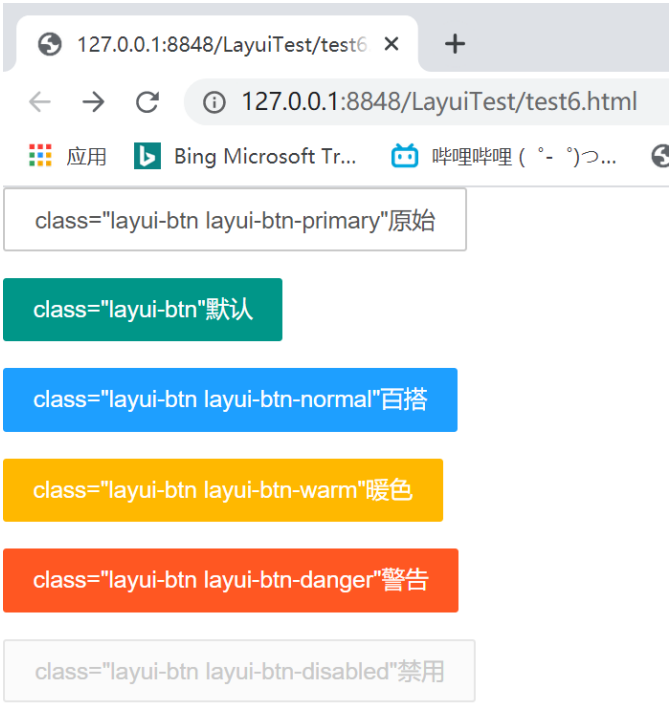
1.7.2 主题

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row">
```

```
            <div class="layui-col-md12">
                <button type="button" class="layui-btn
layui-btn-primary">class="layui-btn layui-btn-primary"原始</button>
                <br /><br />
                <button type="button" class="layui-btn">class="layui-btn"
默认</button>
                <br /><br />
                <button type="button" class="layui-btn
layui-btn-normal">class="layui-btn layui-btn-normal"百搭</button>
                <br /><br />
                <button type="button" class="layui-btn
layui-btn-warm">class="layui-btn layui-btn-warm"暖色</button>
                <br /><br />
                <button type="button" class="layui-btn
layui-btn-danger">class="layui-btn layui-btn-danger"警告</button>
                <br /><br />
                <button type="button" class="layui-btn
layui-btn-disabled">class="layui-btn layui-btn-disabled"禁用</button>
            </div>
        </div>
    </body>
</html>
```

运行效果如图 1-xx 所示。



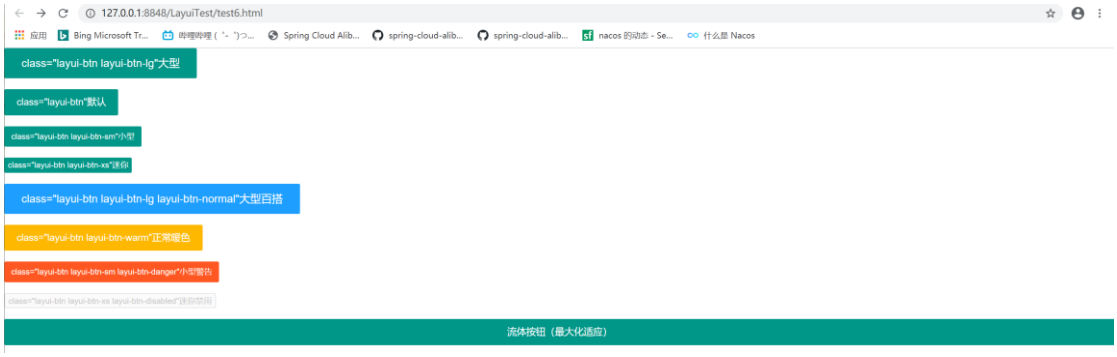
1.7.3 尺寸

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-md12">
                <button type="button" class="layui-btn
layui-btn-lg">class="layui-btn layui-btn-lg"大型</button>
                <br /><br />
                <button type="button" class="layui-btn">class="layui-btn"
默认</button>
                <br /><br />
                <button type="button" class="layui-btn
layui-btn-sm">class="layui-btn layui-btn-sm"小型</button>
                <br /><br />
                <button type="button" class="layui-btn
layui-btn-xs">class="layui-btn layui-btn-xs"迷你</button>
                <br /><br />
            </div>
            <div class="layui-col-md12">
                <button type="button" class="layui-btn layui-btn-lg
layui-btn-normal">class="layui-btn layui-btn-lg
                    layui-btn-normal"大型百搭</button>
                <br /><br />
                <button type="button" class="layui-btn layui-btn-warm">
class="layui-btn layui-btn-warm"正常暖色</button>
                <br /><br />
                <button type="button" class="layui-btn layui-btn-sm
layui-btn-danger">class="layui-btn layui-btn-sm
                    layui-btn-danger"小型警告</button>
                <br /><br />
                <button type="button" class="layui-btn layui-btn-xs
layui-btn-disabled">class="layui-btn layui-btn-xs
                    layui-btn-disabled"迷你禁用</button>
                <br /><br />
```

```
            </div>
            <div class="layui-col-md12">
                <button type="button" class="layui-btn layui-btn-fluid">流体按钮（最大化适应）</button>
            </div>
        </div>
    </body>
</html>
```

运行结果如图 1-xx 所示。



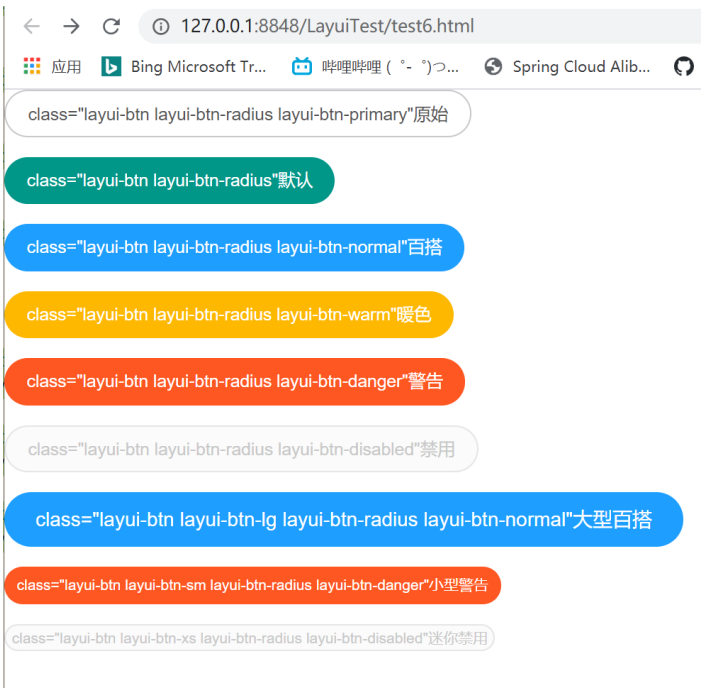
1.7.4 圆角

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-md12">
                <button type="button" class="layui-btn layui-btn-radius layui-btn-primary">class="layui-btn layui-btn-radius
                    layui-btn-primary"原始</button>
                <br /><br />
                <button type="button" class="layui-btn layui-btn-radius">class="layui-btn layui-btn-radius"默认</button>
                <br /><br />
                <button type="button" class="layui-btn layui-btn-radius
```

```
layui-btn-normal">class="layui-btn layui-btn-radius
                    layui-btn-normal"百搭</button>
                <br /><br />
                <button type="button" class="layui-btn layui-btn-radius
layui-btn-warm">class="layui-btn layui-btn-radius
                    layui-btn-warm"暖色</button>
                <br /><br />
                <button type="button" class="layui-btn layui-btn-radius
layui-btn-danger">class="layui-btn layui-btn-radius
                    layui-btn-danger"警告</button>
                <br /><br />
                <button type="button" class="layui-btn layui-btn-radius
layui-btn-disabled">class="layui-btn layui-btn-radius
                    layui-btn-disabled"禁用</button>
                <br /><br />
            </div>
            <div class="layui-col-md12">
                <button type="button" class="layui-btn layui-btn-lg
layui-btn-radius layui-btn-normal">class="layui-btn
                    layui-btn-lg layui-btn-radius layui-btn-normal"大型百搭
</button>
                <br /><br />
                <button type="button" class="layui-btn layui-btn-sm
layui-btn-radius layui-btn-danger">class="layui-btn
                    layui-btn-sm layui-btn-radius layui-btn-danger"小型警告
</button>
                <br /><br />
                <button type="button" class="layui-btn layui-btn-xs
layui-btn-radius layui-btn-disabled">class="layui-btn
                    layui-btn-xs layui-btn-radius layui-btn-disabled"迷你禁
用</button>
            </div>
        </div>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.7.5 图标

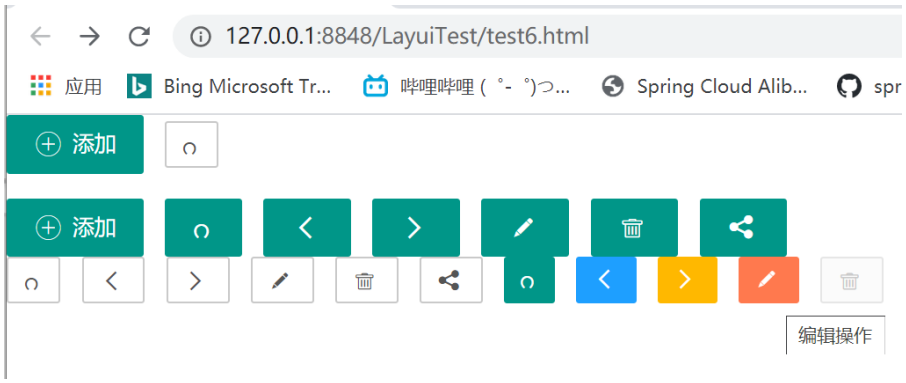
测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-row">
            <div class="layui-col-md12">
                <button type="button" class="layui-btn">
                    <i class="layui-icon">&#xe608;</i> 添加
                </button>
                <button type="button" class="layui-btn layui-btn-sm
layui-btn-primary">
                    <i class="layui-icon">&#x1002;</i>
                </button>
                <br /><br />
            </div>
            <div class="layui-col-md12">
```

```
            <p>
                <button type="button" class="layui-btn"><i
class="layui-icon">&#xe608;</i> 添加</button>
                <button type="button" class="layui-btn"><i
class="layui-icon">&#x1002;</i></button>
                <button type="button" class="layui-btn"><i
class="layui-icon">&#xe603;</i></button>
                <button type="button" class="layui-btn"><i
class="layui-icon">&#xe602;</i></button>
                <button type="button" class="layui-btn"><i
class="layui-icon">&#xe642;</i></button>
                <button type="button" class="layui-btn"><i
class="layui-icon">&#xe640;</i></button>
                <button type="button" class="layui-btn"><i
class="layui-icon">&#xe641;</i></button>
            </p>
            <p>
                <button type="button" class="layui-btn layui-btn-sm
layui-btn-primary"><i class="layui-icon">&#x1002;</i></button>
                <button type="button" class="layui-btn layui-btn-sm
layui-btn-primary"><i class="layui-icon">&#xe603;</i></button>
                <button type="button" class="layui-btn layui-btn-sm
layui-btn-primary"><i class="layui-icon">&#xe602;</i></button>
                <button type="button" class="layui-btn layui-btn-sm
layui-btn-primary"><i class="layui-icon">&#xe642;</i></button>
                <button type="button" class="layui-btn layui-btn-sm
layui-btn-primary"><i class="layui-icon">&#xe640;</i></button>
                <button type="button" class="layui-btn layui-btn-sm
layui-btn-primary"><i class="layui-icon">&#xe641;</i></button>

                <button type="button" class="layui-btn layui-btn-sm"><i
class="layui-icon">&#x1002;</i></button>
                <button type="button" class="layui-btn layui-btn-sm
layui-btn-normal"><i class="layui-icon">&#xe603;</i></button>
                <button type="button" class="layui-btn layui-btn-sm
layui-btn-warm"><i class="layui-icon">&#xe602;</i></button>
                <button type="button" class="layui-btn layui-btn-sm
layui-btn-danger" title="编辑操作"><i
class="layui-icon">&#xe642;</i></button>
                <button type="button" class="layui-btn layui-btn-sm
layui-btn-disabled"><i class="layui-icon">&#xe640;</i></button>
            </p>
        </div>
    </div>
```

```
    </body>
</html>
```

运行结果如图 1-xx 所示。



添加 title="编辑操作"属性使按钮具有提示文本。

1.7.6 按钮组

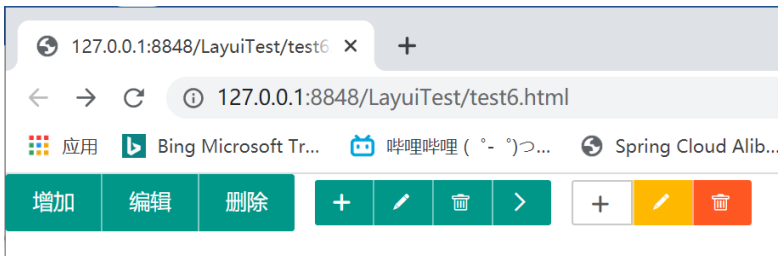
将按钮放入一个 class="layui-btn-group"元素中即可形成按钮组，按钮本身仍然可以随意搭配。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-btn-group">
            <button type="button" class="layui-btn">增加</button>
            <button type="button" class="layui-btn">编辑</button>
            <button type="button" class="layui-btn">删除</button>
        </div>

        <div class="layui-btn-group">
            <button type="button" class="layui-btn layui-btn-sm">
                <i class="layui-icon">&#xe654;</i>
            </button>
            <button type="button" class="layui-btn layui-btn-sm">
                <i class="layui-icon">&#xe642;</i>
            </button>
```

```
            <button type="button" class="layui-btn layui-btn-sm">
                <i class="layui-icon">&#xe640;</i>
            </button>
            <button type="button" class="layui-btn layui-btn-sm">
                <i class="layui-icon">&#xe602;</i>
            </button>
        </div>

        <div class="layui-btn-group">
            <button type="button" class="layui-btn layui-btn-primary
layui-btn-sm">
                <i class="layui-icon">&#xe654;</i>
            </button>
            <button type="button" class="layui-btn layui-btn-warm
layui-btn-sm">
                <i class="layui-icon">&#xe642;</i>
            </button>
            <button type="button" class="layui-btn layui-btn-danger
layui-btn-sm">
                <i class="layui-icon">&#xe640;</i>
            </button>
        </div>
    </body>
</html>
```

运行结果如图 1-xx 所示。



1.7.7 按钮容器

尽管按钮在同节点并排显示时会自动拉开间距，但在按钮太多的情况，效果并不是很好，可以使用按钮容器。

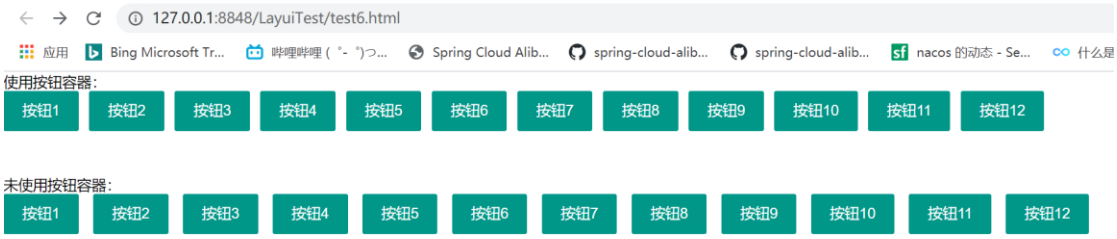
测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
```

```
    <link rel="stylesheet" href="layui/css/layui.css">
    <script src="layui/layui.all.js"></script>
    <script src="js/jquery3.5.1.js"></script>
</head>
<body>
    使用按钮容器:
    <br />
    <div class="layui-row">
        <div class="layui-col-md12">
            <div class="layui-btn-container">
                <button type="button" class="layui-btn">按钮1</button>
                <button type="button" class="layui-btn">按钮2</button>
                <button type="button" class="layui-btn">按钮3</button>
                <button type="button" class="layui-btn">按钮4</button>
                <button type="button" class="layui-btn">按钮5</button>
                <button type="button" class="layui-btn">按钮6</button>
                <button type="button" class="layui-btn">按钮7</button>
                <button type="button" class="layui-btn">按钮8</button>
                <button type="button" class="layui-btn">按钮9</button>
                <button type="button" class="layui-btn">按钮10</button>
                <button type="button" class="layui-btn">按钮11</button>
                <button type="button" class="layui-btn">按钮12</button>
            </div>
        </div>
    </div>
    <br />
    <br />
    未使用按钮容器:
    <br />
    <div class="layui-row">
        <div class="layui-col-md12">
            <button type="button" class="layui-btn">按钮1</button>
            <button type="button" class="layui-btn">按钮2</button>
            <button type="button" class="layui-btn">按钮3</button>
            <button type="button" class="layui-btn">按钮4</button>
            <button type="button" class="layui-btn">按钮5</button>
            <button type="button" class="layui-btn">按钮6</button>
            <button type="button" class="layui-btn">按钮7</button>
            <button type="button" class="layui-btn">按钮8</button>
            <button type="button" class="layui-btn">按钮9</button>
            <button type="button" class="layui-btn">按钮10</button>
            <button type="button" class="layui-btn">按钮11</button>
            <button type="button" class="layui-btn">按钮12</button>
        </div>
```

```
        </div>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.8 表单

在一个容器中设置 class="layui-form"来标识一个表单元素块，通过规范好的 HTML 结构及 CSS 类来组装成各式各样的表单元素，并通过内置的 form 模块来完成各种交互。

必须要加载 form 模块。请注意：如果使用 layui.js 并且不加载 form 模块，则 select、checkbox、radio 等组件将无法显示，呈隐藏状态，并且无法使用 form 相关功能，可以使用如下代码加载 form 模块：

```
    <script>
        layui.use('form', function() {
            var form = layui.form; //只有执行了这一步，部分表单元素才会自动修饰成功
            //但是，如果HTML元素是动态生成的，自动渲染就会失效
            //因此需要在相应的地方，执行下述方法来进行渲染
            form.render();
        });
    </script>
```

如果使用layui.all.js，则不需要use()方法：

```
    <script>
        var form = layui.form;
        form.render();
    </script>
```

<script></script>此段 javascript 代码建议放在页面的最后。

1.8.1 小睹为快

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
```

```
        <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <form class="layui-form" action="">
            <div class="layui-form-item">
                <label class="layui-form-label">输入框</label>
                <div class="layui-input-block">
                    <input type="text" name="title" required
lay-verify="required" placeholder="请输入标题" autocomplete="off"
class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">密码框</label>
                <div class="layui-input-inline">
                    <input type="password" name="password" required
lay-verify="required" placeholder="请输入密码" autocomplete="off"
class="layui-input">
                </div>
                <div class="layui-form-mid layui-word-aux">辅助文字</div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">选择框</label>
                <div class="layui-input-block">
                    <select name="city" lay-verify="required">
                        <option value=""></option>
                        <option value="0">北京</option>
                        <option value="1">上海</option>
                        <option value="2">广州</option>
                        <option value="3">深圳</option>
                        <option value="4">杭州</option>
                    </select>
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">复选框</label>
                <div class="layui-input-block">
                    <input type="checkbox" name="like[write]" title="写作">
                    <input type="checkbox" name="like[read]" title="阅读"
checked>
```

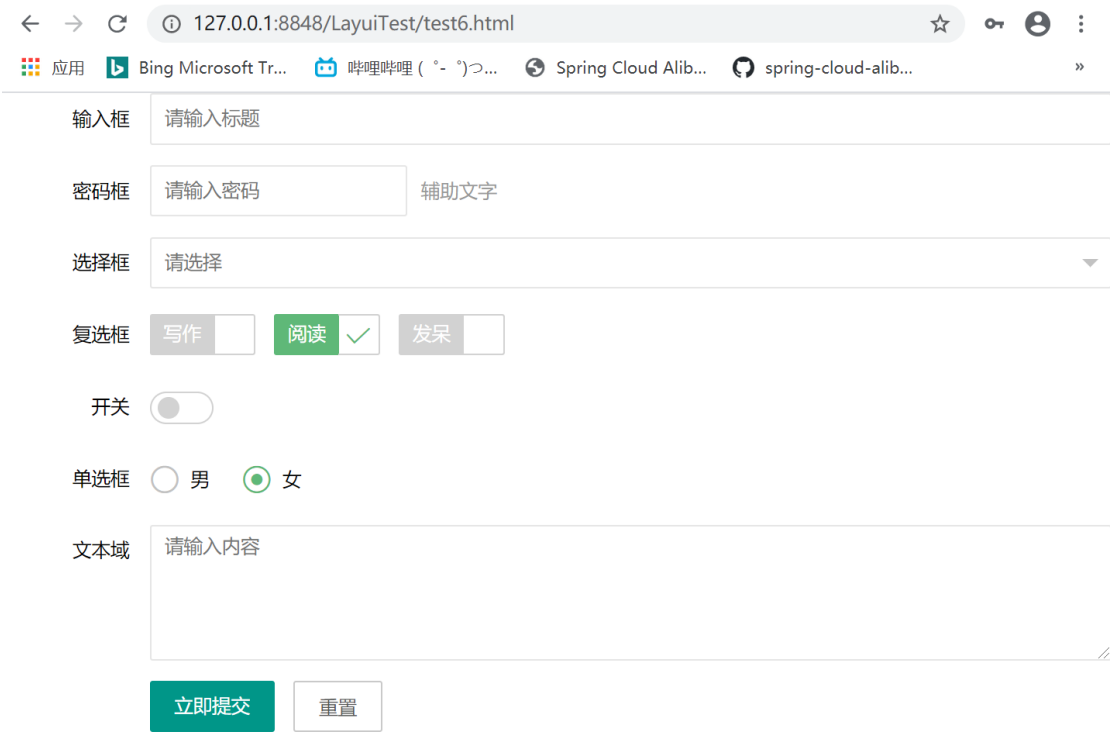
```
                <input type="checkbox" name="like[dai]" title="发呆">
            </div>
        </div>
        <div class="layui-form-item">
            <label class="layui-form-label">开关</label>
            <div class="layui-input-block">
                <input type="checkbox" name="switch" lay-skin="switch">
            </div>
        </div>
        <div class="layui-form-item">
            <label class="layui-form-label">单选框</label>
            <div class="layui-input-block">
                <input type="radio" name="sex" value="男" title="男">
                <input type="radio" name="sex" value="女" title="女"
checked>
            </div>
        </div>
        <div class="layui-form-item layui-form-text">
            <label class="layui-form-label">文本域</label>
            <div class="layui-input-block">
                <textarea name="desc" placeholder="请输入内容"
class="layui-textarea"></textarea>
            </div>
        </div>
        <div class="layui-form-item">
            <div class="layui-input-block">
                <button class="layui-btn" lay-submit
lay-filter="formDemo">立即提交</button>
                <button type="reset" class="layui-btn
layui-btn-primary">重置</button>
            </div>
        </div>
    </form>
    <script src="layui/layui.all.js"></script>
    <script>
        var form = layui.form; //只有执行了这一步，部分表单元素才会自动修
饰成功
        //但是，如果你的HTML是动态生成的，自动渲染就会失效
        //因此你需要在相应的地方，执行下述方法来进行渲染
        form.render();

        form.on('submit(formDemo)', function(data) {
            layer.msg(JSON.stringify(data.field));
            return false;
```

```
        });
    </script>
</body>
</html>
```

基本的表单行区域由下面的代码模块组成：

```
<div class="layui-form-item">
    <label class="layui-form-label">标签区域</label>
    <div class="layui-input-block">原始表单元素区域</div>
</div>
```

运行结果如图 1-xx 所示。



1.8.2 输入框

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="layui/layui.all.js"></script>
        <script src="js/jquery3.5.1.js"></script>
```

```
    </head>
    <body>
        <form class="layui-form" action="">
            <div class="layui-form-item">
                <label class="layui-form-label">账号</label>
                <div class="layui-input-block">
                    <input type="text" name="username" required
lay-verify="required" placeholder="请输入账号" autocomplete="off"
class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">密码</label>
                <div class="layui-input-block">
                    <input type="password" name="password" required
lay-verify="required" placeholder="请输入密码" autocomplete="off"
class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">电话</label>
                <div class="layui-input-block">
                    <input type="text" name="phone" required
lay-verify="phone" placeholder="请输入电话" autocomplete="off"
class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">邮箱</label>
                <div class="layui-input-block">
                    <input type="text" name="email" required
lay-verify="email" placeholder="请输入邮箱" autocomplete="off"
class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">网址</label>
                <div class="layui-input-block">
                    <input type="text" name="url" required lay-verify="url"
placeholder="请输入网址" autocomplete="off" class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">身高</label>
```

```
                <div class="layui-input-block">
                    <input type="text" name="height" required
lay-verify="number" placeholder="请输入身高" autocomplete="off"
class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">日期</label>
                <div class="layui-input-block">
                    <input type="text" name="insertdate" required
lay-verify="date" placeholder="请输入日期" autocomplete="off"
class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">身份证号</label>
                <div class="layui-input-block">
                    <input type="text" name="userId" required
lay-verify="identity" placeholder="请输入身份证号" autocomplete="off"
class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <div class="layui-input-block">
                    <button class="layui-btn" lay-submit
lay-filter="formDemo">立即提交</button>
                    <button type="reset" class="layui-btn
layui-btn-primary">重置</button>
                </div>
            </div>
        </form>
        <script src="layui/layui.all.js"></script>
        <script>
            var form = layui.form; //只有执行了这一步，部分表单元素才会自动修
饰成功
            //但是，如果你的HTML是动态生成的，自动渲染就会失效
            //因此你需要在相应的地方，执行下述方法来进行渲染
            form.render();

            form.on('submit(formDemo)', function(data) {
                layer.msg(JSON.stringify(data.field));
                return false;
            });
        </script>
```

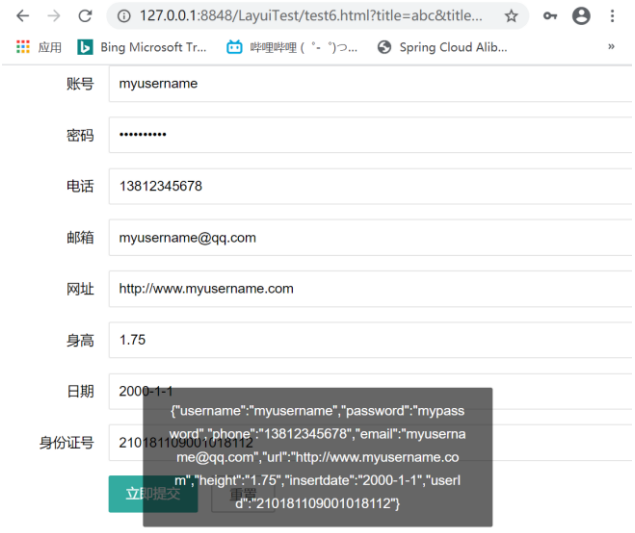
```
    </body>
</html>
```

required：必填字段。

lay-verify：需要验证的类型。required 必填项，phone 手机号，email 邮箱地址，url 网址，number 数字，date 日期，identity 身份证号。

class="layui-input"：layui 提供的通用 CSS 类。

运行结果如图 1-xx 所示。



1.8.3 下拉选择框

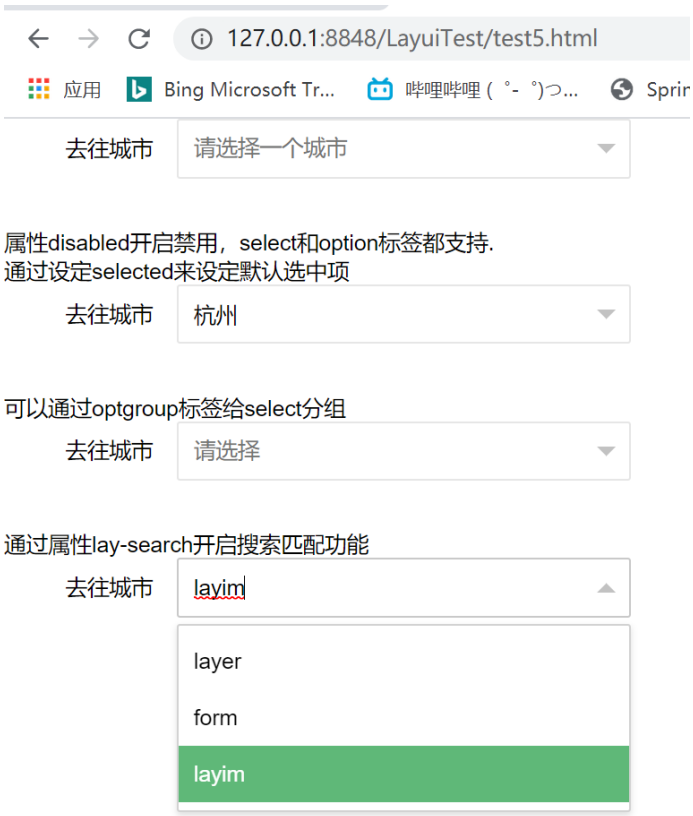
测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-form" style="width: 400px;">
            <div class="layui-form-item">
                <label class="layui-form-label">去往城市</label>
                <div class="layui-input-block">
                    <select name="city" lay-verify="">
                        <option value="">请选择一个城市</option>
```

```
                    <option value="010">北京</option>
                    <option value="021">上海</option>
                    <option value="0571">杭州</option>
                </select>
            </div>
        </div>
        <br />
        属性disabled开启禁用，select和option标签都支持.<br />
        通过设置selected来设置默认选中项<br />
        <div class="layui-form-item">
            <label class="layui-form-label">去往城市</label>
            <div class="layui-input-block">
                <select name="city" lay-verify="">
                    <option value="010">北京</option>
                    <option value="021" disabled>上海(禁用效果)</option>
                    <option value="0571" selected>杭州</option>
                </select>
            </div>
        </div>
        <br />
        可以通过optgroup标签给select分组<br />
        <div class="layui-form-item">
            <label class="layui-form-label">去往城市</label>
            <div class="layui-input-block">
                <select name="quiz">
                    <option value="">请选择</option>
                    <optgroup label="城市记忆">
                        <option value="你工作的第一个城市">你工作的第一个
城市？</option>
                    </optgroup>
                    <optgroup label="学生时代">
                        <option value="你的工号">你的工号？</option>
                        <option value="你最喜欢的老师">你最喜欢的老师？
</option>
                    </optgroup>
                </select>
            </div>
        </div>
        <br />
        通过属性lay-search开启搜索匹配功能<br />
        <div class="layui-form-item">
            <label class="layui-form-label">去往城市</label>
            <div class="layui-input-block">
                <select name="city" lay-verify="" lay-search>
```

```
                    <option value="010">layer</option>
                    <option value="021">form</option>
                    <option value="0571" selected>layim</option>
                </select>
            </div>
        </div>
    </div>
    <script src="layui/layui.all.js"></script>
    <script>
        var form = layui.form; //只有执行了这一步，部分表单元素才会自动修饰成功
        //但是，如果你的HTML是动态生成的，自动渲染就会失效
        //因此你需要在相应的地方，执行下述方法来进行渲染
        form.render();
    </script>
</body>
</html>
```

运行结果如图 1-xx 所示。



1.8.4 复选框

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <form class="layui-form" action="">
            默认风格：<br />
            <div class="layui-form-item">
                <div class="layui-input-block">
                    <input type="checkbox" name="" title="写作" checked>
                </div>
            </div>
            <div class="layui-form-item">
                <div class="layui-input-block">
                    <input type="checkbox" name="" title="发呆">
                </div>
            </div>
            <div class="layui-form-item">
                <div class="layui-input-block">
                    <input type="checkbox" name="" title="禁用" disabled>
                </div>
            </div>
            <br />
            原始风格：<br />
            <div class="layui-form-item">
                <div class="layui-input-block">
                    <input type="checkbox" name="" title="写作"
lay-skin="primary" checked>
                </div>
            </div>
            <div class="layui-form-item">
                <div class="layui-input-block">
                    <input type="checkbox" name="" title="发呆"
lay-skin="primary">
                </div>
```

```
            </div>
            <div class="layui-form-item">
                <div class="layui-input-block">
                    <input type="checkbox" name="" title="禁用"
lay-skin="primary" disabled>
                </div>
            </div>
        </form>
        <script src="layui/layui.all.js"></script>
        <script>
            var form = layui.form; //只有执行了这一步，部分表单元素才会自动修
饰成功
            //但是，如果你的HTML是动态生成的，自动渲染就会失效
            //因此你需要在相应的地方，执行下述方法来进行渲染
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。

属性 title 可自定义文本。如果只想显示复选框，可以不用设置 title 属性。
属性 checked 可设置默认选中。
属性 lay-skin 可设置复选框的风格。
设置 value="1"可自定义值，否则是默认值 on。

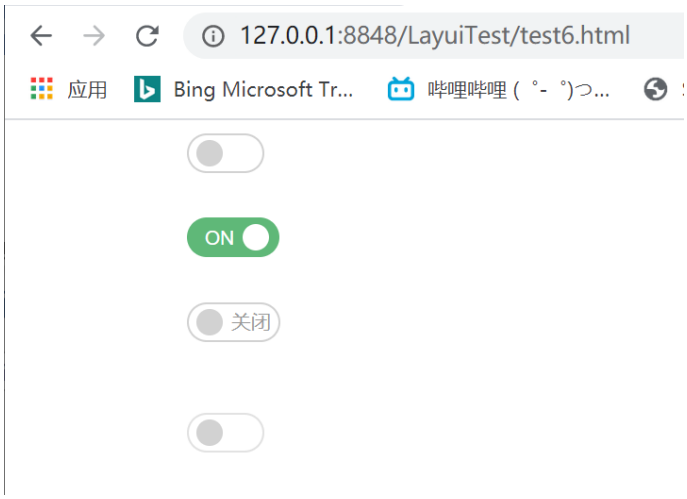
1.8.5 开关

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <form class="layui-form" action="">
            <div class="layui-form-item">
                <div class="layui-input-block">
                    <input type="checkbox" name="xxx" lay-skin="switch">
                </div>
            </div>
            <div class="layui-form-item">
                <div class="layui-input-block">
                    <input type="checkbox" name="yyy" lay-skin="switch"
lay-text="ON|OFF" checked>
                </div>
            </div>
            <div class="layui-form-item">
                <div class="layui-input-block">
                    <input type="checkbox" name="zzz" lay-skin="switch"
lay-text="开启|关闭">
                </div>
            </div>
            <br />
            <div class="layui-form-item">
                <div class="layui-input-block">
                    <input type="checkbox" name="aaa" lay-skin="switch"
disabled>
                </div>
            </div>
        </form>
        <script src="layui/layui.all.js"></script>
        <script>
            var form = layui.form; //只有执行了这一步，部分表单元素才会自动修
饰成功
```

```
            //但是，如果你的HTML是动态生成的，自动渲染就会失效
            //因此你需要在相应的地方，执行下述方法来进行渲染
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



开关是 checkbox 复选框的“变种”，通过设置 lay-skin="switch"形成了开关风格。
属性 checked 可设置默认开。
属性 disabled 开启禁用。
属性 lay-text 可自定义开关两种状态的文本。
设置 value="1"可自定义值，否则是默认值 on。

1.8.6 单选框

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <form class="layui-form" action="">
            <div class="layui-form-item">
                <div class="layui-input-block">
```

```
                <input type="radio" name="sex" value="nan" title="男">
            </div>
        </div>
        <div class="layui-form-item">
            <div class="layui-input-block">
                <input type="radio" name="sex" value="nv" title="女"
checked>
            </div>
        </div>
        <div class="layui-form-item">
            <div class="layui-input-block">
                <input type="radio" name="sex" value="" title="中性"
disabled>
            </div>
        </div>
    </form>
    <script src="layui/layui.all.js"></script>
    <script>
        var form = layui.form; //只有执行了这一步，部分表单元素才会自动修
饰成功
        //但是，如果你的HTML是动态生成的，自动渲染就会失效
        //因此你需要在相应的地方，执行下述方法来进行渲染
        form.render();
    </script>
</body>
</html>
```

运行结果如图 1-xx 所示。

属性 title 可自定义文本。
属性 disabled 开启禁用。
设置 value="1"可自定义值，否则是默认值 on。

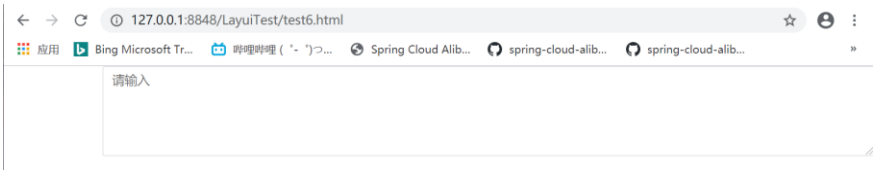
1.8.7 文本域

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <form class="layui-form" action="">
            <div class="layui-form-item">
                <div class="layui-input-block">
                    <textarea name="" required lay-verify="required" placeholder="请输入" class="layui-textarea"></textarea>
                </div>
            </div>
        </form>
        <script src="layui/layui.all.js"></script>
        <script>
            var form = layui.form; //只有执行了这一步，部分表单元素才会自动修饰成功
            //但是，如果你的HTML是动态生成的，自动渲染就会失效
            //因此你需要在相应的地方，执行下述方法来进行渲染
            form.render();
        </script>
    </body>
</html>
```

class="layui-textarea"是 layui 提供的通用 CSS 类。

运行结果如图 1-xx 所示。



1.8.8 组装行内表单

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-form-item">
            <div class="layui-inline">
                <label class="layui-form-label">范围</label>
                <div class="layui-input-inline" style="width: 100px;">
                    <input type="text" name="price_min" placeholder="￥"
autocomplete="off" class="layui-input">
                </div>
                <div class="layui-form-mid">-</div>
                <div class="layui-input-inline" style="width: 100px;">
                    <input type="text" name="price_max" placeholder="￥"
autocomplete="off" class="layui-input">
                </div>
            </div>

            <div class="layui-inline">
                <label class="layui-form-label">密码</label>
                <div class="layui-input-inline" style="width: 100px;">
                    <input type="password" name="" autocomplete="off"
class="layui-input">
                </div>
            </div>
        </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var form = layui.form; //只有执行了这一步，部分表单元素才会自动修
饰成功
            //但是，如果你的HTML是动态生成的，自动渲染就会失效
            //因此你需要在相应的地方，执行下述方法来进行渲染
            form.render();
        </script>
```

```
    </body>
</html>
```

运行结果如图 1-xx 所示。

class="layui-inline"：定义外层行内。
class="layui-input-inline"：定义内层行内。

1.8.9 忽略美化渲染

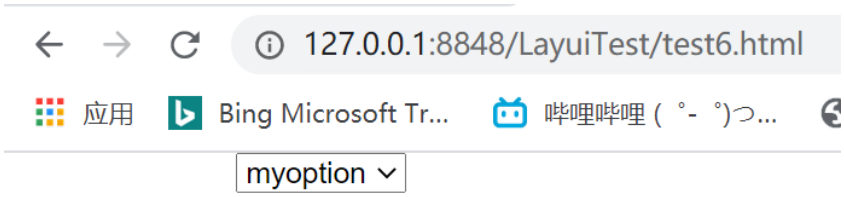
可以对表单元素增加属性 lay-ignore，将不会对该标签进行美化渲染，即保留系统风格。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <form class="layui-form" action="">
            <div class="layui-form-item">
                <div class="layui-input-block">
                    <select lay-ignore>
                        <option>myoption</option>
                    </select>
                </div>
            </div>
        </form>
        <script src="layui/layui.all.js"></script>
        <script>
            var form = layui.form; //只有执行了这一步，部分表单元素才会自动修
饰成功
            //但是，如果你的HTML是动态生成的，自动渲染就会失效
```

```
            //因此你需要在相应的地方，执行下述方法来进行渲染
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



一般不推荐这样做。

1.8.10 表单方框风格

通过追加 layui-form-pane 的 class 来设置表单为方框风格，但内部结构不变。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        内部结构都一样，值得注意的是 复选框/开关/单选框 这些组合在该风格下需要额外添加 pane属性（否则会看起来比较别扭）。
        <br />
        <br />
        <form class="layui-form layui-form-pane" action="">
            <div class="layui-form-item" pane>
                <label class="layui-form-label">文本</label>
                <div class="layui-input-block">
                    <input type="text" name="title" required lay-verify="required" placeholder="请输入标题" autocomplete="off" class="layui-input">
                </div>
```

```
            </div>
            <div class="layui-form-item" pane>
                <label class="layui-form-label">密码</label>
                <div class="layui-input-block">
                    <input type="password" name="title" required
lay-verify="required" placeholder="请输入密码" autocomplete="off"
class="layui-input">
                </div>
            </div>
            <div class="layui-form-item" pane>
                <label class="layui-form-label">下拉框</label>
                <div class="layui-input-block">
                    <select name="city" lay-verify="">
                        <option value="">请选择一个城市</option>
                        <option value="010">北京</option>
                        <option value="021">上海</option>
                        <option value="0571">杭州</option>
                    </select>

                </div>
            </div>
            <div class="layui-form-item" pane>
                <label class="layui-form-label">复选框</label>
                <div class="layui-input-block">
                    <input type="checkbox" name="" title="旅游"
lay-skin="primary" checked>
                    <input type="checkbox" name="" title="游戏"
lay-skin="primary" checked>
                    <input type="checkbox" name="" title="动漫"
lay-skin="primary" checked>
                    <input type="checkbox" name="" title="B站"
lay-skin="primary" checked>
                </div>
            </div>
            <div class="layui-form-item" pane>
                <label class="layui-form-label">单选框</label>
                <div class="layui-input-block">
                    <input type="radio" name="sex" value="男" title="男">
                    <input type="radio" name="sex" value="女" title="女"
checked>
                </div>
            </div>
            <div class="layui-form-item" pane>
                <label class="layui-form-label">文本域</label>
```

```
                <div class="layui-input-block">
                    <textarea name="" required lay-verify="required"
placeholder="请输入" class="layui-textarea"></textarea>
                </div>
            </div>
        </form>
        <script src="layui/layui.all.js"></script>
        <script>
            var form = layui.form; //只有执行了这一步，部分表单元素才会自动修
饰成功
            //但是，如果你的HTML是动态生成的，自动渲染就会失效
            //因此你需要在相应的地方，执行下述方法来进行渲染
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。

1.9 导航

导航一般指页面引导性频道集合，多以菜单的形式呈现，可应用于头部和侧边。面包屑结构简单，支持自定义分隔符。

注意：千万不要忘了加载 element 模块。虽然大部分行为都是在加载完该模块后自动完成的，但一些交互操作，如弹出二级菜单等，需借助 element 模块才能使用。

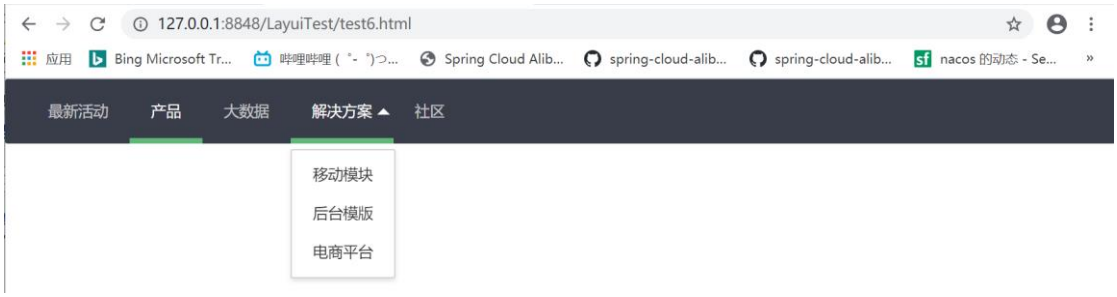
1.9.1 水平导航

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <ul class="layui-nav" lay-filter="">
            <li class="layui-nav-item"><a href="">最新活动</a></li>
            <li class="layui-nav-item layui-this"><a href="">产品</a></li>
            <li class="layui-nav-item"><a href="">大数据</a></li>
            <li class="layui-nav-item">
                <a href="javascript:;">解决方案</a>
                <dl class="layui-nav-child">
                    <!-- 二级菜单 -->
```

```
                <dd><a href="">移动模块</a></dd>
                <dd><a href="">后台模版</a></dd>
                <dd><a href="">电商平台</a></dd>
            </dl>
        </li>
        <li class="layui-nav-item"><a href="">社区</a></li>
    </ul>
    <script src="layui/layui.all.js"></script>
    <script>
        var element = layui.element;
        var form = layui.form;
        form.render();
    </script>
</body>
</html>
```

运行结果如图 1-xx 所示。



设置 layui-this 来实现默认选中。

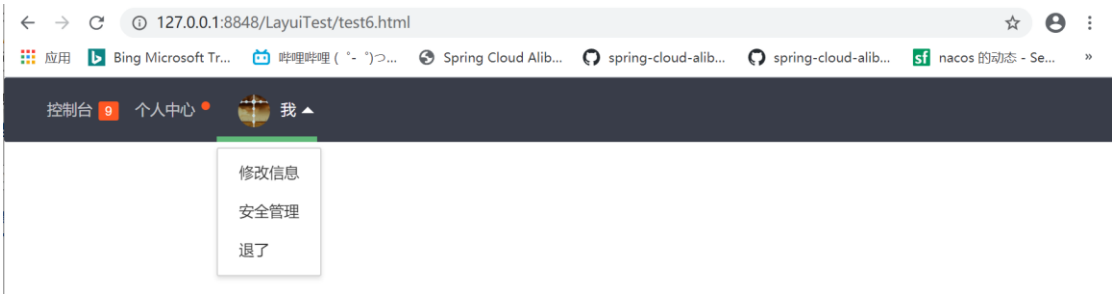
除了一般的文字导航，还内置了图片和徽章的支持。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <ul class="layui-nav">
            <li class="layui-nav-item">
                <a href="">控制台<span class="layui-badge">9</span></a>
            </li>
            <li class="layui-nav-item">
```

```
            <a href="">个人中心<span
class="layui-badge-dot"></span></a>
        </li>
        <li class="layui-nav-item">
            <a href=""><img src="//t.cn/RCzsdCq" class="layui-nav-img">
我</a>
            <dl class="layui-nav-child">
                <dd><a href="javascript:;">修改信息</a></dd>
                <dd><a href="javascript:;">安全管理</a></dd>
                <dd><a href="javascript:;">退了</a></dd>
            </dl>
        </li>
    </ul>
    <script src="layui/layui.all.js"></script>
    <script>
        var element = layui.element;
        var form = layui.form;
        form.render();
    </script>
</body>
</html>
```

运行结果如图 1-xx 所示。



1.9.2 导航主题

通过对导航追加 CSS 背景类，让导航呈现不同的主题色。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
```

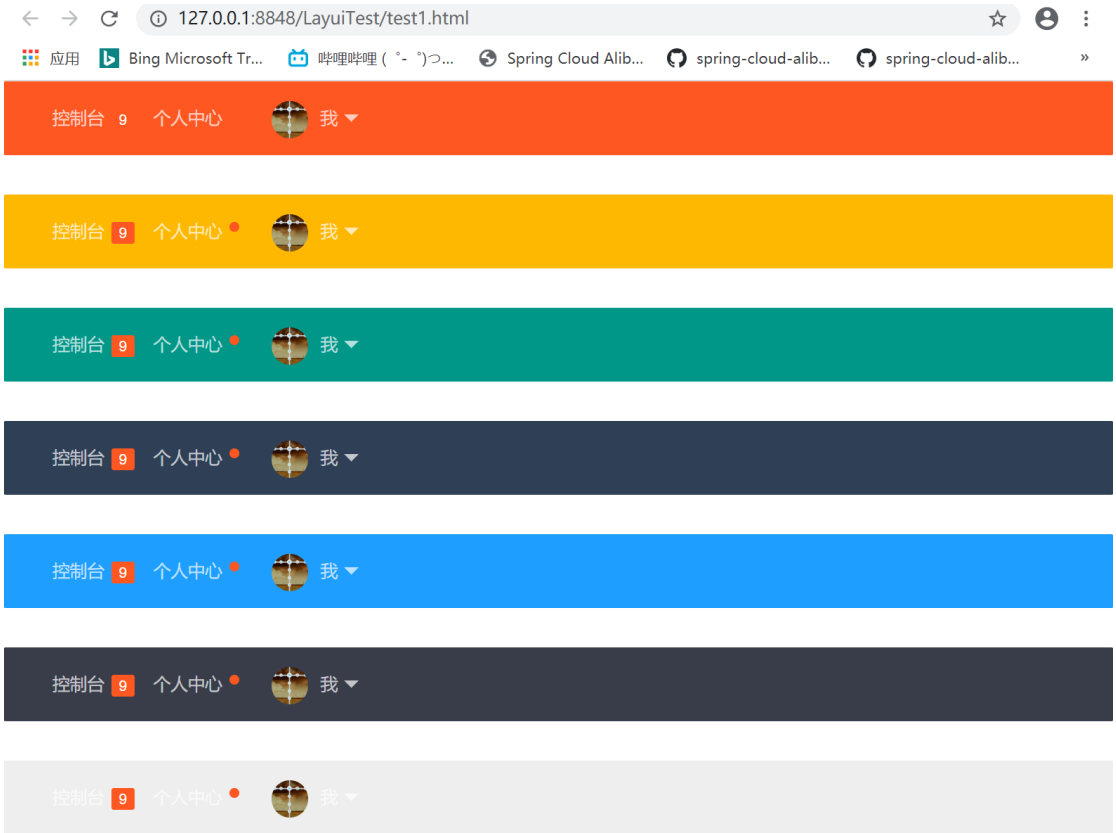
```
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <ul class="layui-nav layui-bg-red">
            <li class="layui-nav-item">
                <a href="">控制台<span class="layui-badge">9</span></a>
            </li>
            <li class="layui-nav-item">
                <a href="">个人中心<span
class="layui-badge-dot"></span></a>
            </li>
            <li class="layui-nav-item">
                <a href=""><img src="//t.cn/RCzsdCq" class="layui-nav-img">
我</a>
                <dl class="layui-nav-child">
                    <dd><a href="javascript:;">修改信息</a></dd>
                    <dd><a href="javascript:;">安全管理</a></dd>
                    <dd><a href="javascript:;">退了</a></dd>
                </dl>
            </li>
        </ul>
        <br />
        <br />
        <ul class="layui-nav layui-bg-orange">
            <li class="layui-nav-item">
                <a href="">控制台<span class="layui-badge">9</span></a>
            </li>
            <li class="layui-nav-item">
                <a href="">个人中心<span
class="layui-badge-dot"></span></a>
            </li>
            <li class="layui-nav-item">
                <a href=""><img src="//t.cn/RCzsdCq" class="layui-nav-img">
我</a>
                <dl class="layui-nav-child">
                    <dd><a href="javascript:;">修改信息</a></dd>
                    <dd><a href="javascript:;">安全管理</a></dd>
                    <dd><a href="javascript:;">退了</a></dd>
                </dl>
            </li>
        </ul>
        <br />
        <br />
```

```
        <ul class="layui-nav layui-bg-green">
            <li class="layui-nav-item">
                <a href="">控制台<span class="layui-badge">9</span></a>
            </li>
            <li class="layui-nav-item">
                <a href="">个人中心<span
class="layui-badge-dot"></span></a>
            </li>
            <li class="layui-nav-item">
                <a href=""><img src="//t.cn/RCzsdCq" class="layui-nav-img">
我</a>
                <dl class="layui-nav-child">
                    <dd><a href="javascript:;">修改信息</a></dd>
                    <dd><a href="javascript:;">安全管理</a></dd>
                    <dd><a href="javascript:;">退了</a></dd>
                </dl>
            </li>
        </ul>
        <br />
        <br />
        <ul class="layui-nav layui-bg-cyan">
            <li class="layui-nav-item">
                <a href="">控制台<span class="layui-badge">9</span></a>
            </li>
            <li class="layui-nav-item">
                <a href="">个人中心<span
class="layui-badge-dot"></span></a>
            </li>
            <li class="layui-nav-item">
                <a href=""><img src="//t.cn/RCzsdCq" class="layui-nav-img">
我</a>
                <dl class="layui-nav-child">
                    <dd><a href="javascript:;">修改信息</a></dd>
                    <dd><a href="javascript:;">安全管理</a></dd>
                    <dd><a href="javascript:;">退了</a></dd>
                </dl>
            </li>
        </ul>
        <br />
        <br />
        <ul class="layui-nav layui-bg-blue">
            <li class="layui-nav-item">
                <a href="">控制台<span class="layui-badge">9</span></a>
            </li>
```

```
            <li class="layui-nav-item">
                <a href="">个人中心<span
class="layui-badge-dot"></span></a>
            </li>
            <li class="layui-nav-item">
                <a href=""><img src="//t.cn/RCzsdCq" class="layui-nav-img">
我</a>
                <dl class="layui-nav-child">
                    <dd><a href="javascript:;">修改信息</a></dd>
                    <dd><a href="javascript:;">安全管理</a></dd>
                    <dd><a href="javascript:;">退了</a></dd>
                </dl>
            </li>
        </ul>
        <br />
        <br />
        <ul class="layui-nav layui-bg-black">
            <li class="layui-nav-item">
                <a href="">控制台<span class="layui-badge">9</span></a>
            </li>
            <li class="layui-nav-item">
                <a href="">个人中心<span
class="layui-badge-dot"></span></a>
            </li>
            <li class="layui-nav-item">
                <a href=""><img src="//t.cn/RCzsdCq" class="layui-nav-img">
我</a>
                <dl class="layui-nav-child">
                    <dd><a href="javascript:;">修改信息</a></dd>
                    <dd><a href="javascript:;">安全管理</a></dd>
                    <dd><a href="javascript:;">退了</a></dd>
                </dl>
            </li>
        </ul>
        <br />
        <br />
        <ul class="layui-nav layui-bg-gray">
            <li class="layui-nav-item">
                <a href="">控制台<span class="layui-badge">9</span></a>
            </li>
            <li class="layui-nav-item">
                <a href="">个人中心<span
class="layui-badge-dot"></span></a>
            </li>
```

```
            <li class="layui-nav-item">
                <a href=""><img src="//t.cn/RCzsdCq" class="layui-nav-img">
我</a>
                <dl class="layui-nav-child">
                    <dd><a href="javascript:;">修改信息</a></dd>
                    <dd><a href="javascript:;">安全管理</a></dd>
                    <dd><a href="javascript:;">退了</a></dd>
                </dl>
            </li>
        </ul>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            var form = layui.form;
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



1.9.3 垂直导航

水平、垂直、侧边三个导航的 HTML 结构是完全一样的，不同的是：

（1）垂直导航 class 需要追加 layui-nav-tree

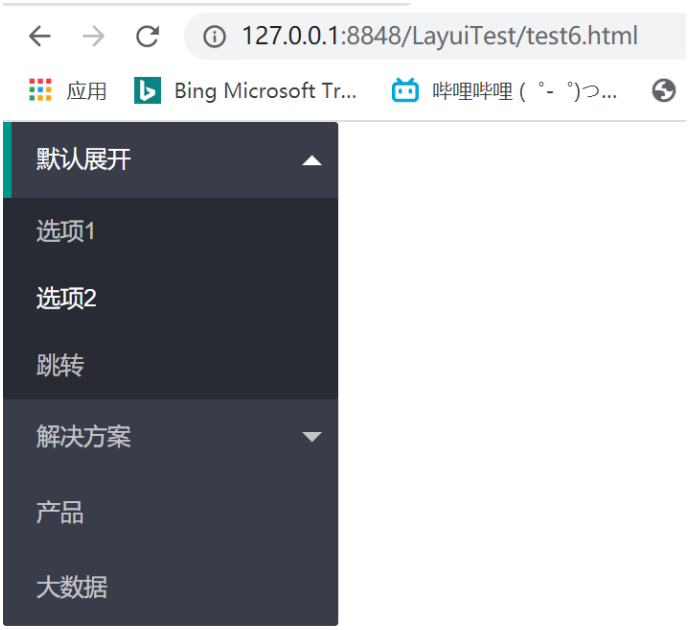
（2）侧边导航 class 需要追加 layui-nav-tree layui-nav-side

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <ul class="layui-nav layui-nav-tree" lay-filter="test">
            <li class="layui-nav-item layui-nav-itemed">
                <a href="javascript:;">默认展开</a>
                <dl class="layui-nav-child">
                    <dd><a href="javascript:;">选项1</a></dd>
                    <dd><a href="javascript:;">选项2</a></dd>
                    <dd><a href="">跳转</a></dd>
                </dl>
            </li>
            <li class="layui-nav-item">
                <a href="javascript:;">解决方案</a>
                <dl class="layui-nav-child">
                    <dd><a href="">移动模块</a></dd>
                    <dd><a href="">后台模版</a></dd>
                    <dd><a href="">电商平台</a></dd>
                </dl>
            </li>
            <li class="layui-nav-item"><a href="">产品</a></li>
            <li class="layui-nav-item"><a href="">大数据</a></li>
        </ul>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            var form = layui.form;
            form.render();
        </script>
```

```
    </body>
</html>
```

运行结果如图 1-xx 所示。



1.9.4 侧边导航

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <ul class="layui-nav layui-nav-tree layui-nav-side">
            <li class="layui-nav-item layui-nav-itemed">
                <a href="javascript:;">默认展开</a>
                <dl class="layui-nav-child">
                    <dd><a href="javascript:;">选项1</a></dd>
                    <dd><a href="javascript:;">选项2</a></dd>
                    <dd><a href="">跳转</a></dd>
```

```
            </dl>
        </li>
        <li class="layui-nav-item">
            <a href="javascript:;">解决方案</a>
            <dl class="layui-nav-child">
                <dd><a href="">移动模块</a></dd>
                <dd><a href="">后台模版</a></dd>
                <dd><a href="">电商平台</a></dd>
            </dl>
        </li>
        <li class="layui-nav-item"><a href="">产品</a></li>
        <li class="layui-nav-item"><a href="">大数据</a></li>
    </ul>
    <script src="layui/layui.all.js"></script>
    <script>
        var element = layui.element;
        var form = layui.form;
        form.render();
    </script>
</body>
</html>
```

运行结果如图 1-xx 所示。

1.9.5 导航可选属性

导航可选属性如图 1-xx 所示。

对导航元素结构设定可选属性，可让导航菜单项达到不同效果。目前支持的属性如下：

属性名	可选值	说明
lay-shrink	• *空值*（默认） 不收缩兄弟菜单子菜单 • *all* 收缩全部兄弟菜单子菜单	展开子菜单时，是否收缩兄弟节点已展开的子菜单（注：layui 2.2.6 开始新增） 如：*<ul class="layui-nav layui-nav-tree" lay-shrink="all"> … *
lay-unselect	无需填值	点击指定导航菜单时，不会出现选中效果（注：layui 2.2.0 开始新增） 如：*<li class="layui-nav-item" lay-unselect>刷新 *

1.9.5.1 测试属性 lay-shrink

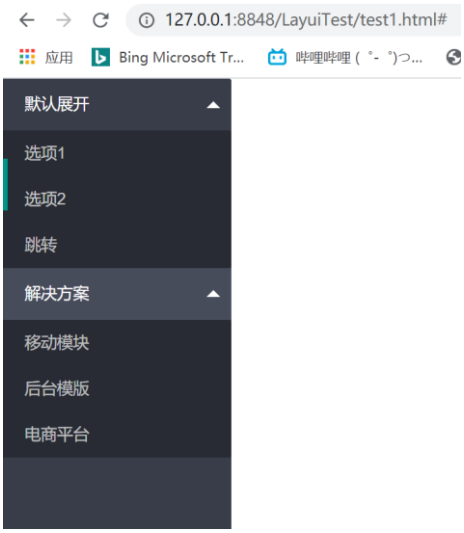
无 lay-shrink="all"属性可以同时展开多个菜单。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <ul class="layui-nav layui-nav-tree layui-nav-side">
            <li class="layui-nav-item layui-nav-itemed">
                <a href="javascript:;">默认展开</a>
                <dl class="layui-nav-child">
                    <dd><a href="javascript:;">选项1</a></dd>
                    <dd><a href="javascript:;">选项2</a></dd>
                    <dd><a href="">跳转</a></dd>
                </dl>
            </li>
            <li class="layui-nav-item">
                <a href="javascript:;">解决方案</a>
                <dl class="layui-nav-child">
                    <dd><a href="">移动模块</a></dd>
                    <dd><a href="">后台模版</a></dd>
                    <dd><a href="">电商平台</a></dd>
                </dl>
```

```
            </li>
        </ul>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            var form = layui.form;
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



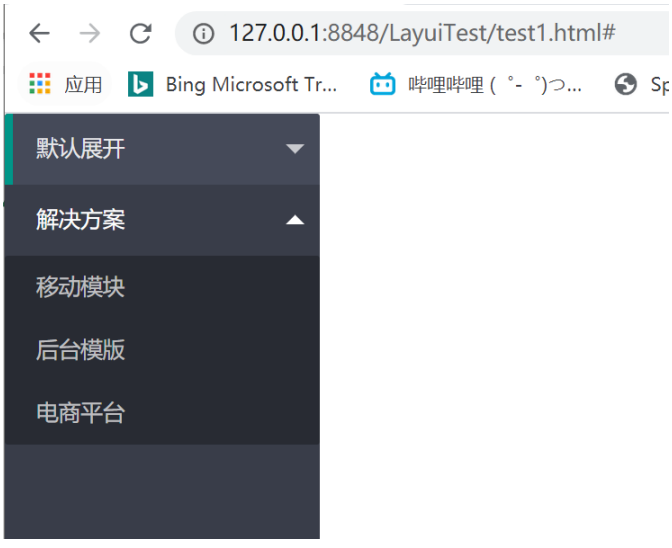
有 lay-shrink="all"属性同时只能展开 1 个菜单。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <ul class="layui-nav layui-nav-tree layui-nav-side"
lay-shrink="all">
            <li class="layui-nav-item layui-nav-itemed">
                <a href="javascript:;">默认展开</a>
                <dl class="layui-nav-child">
```

```
                <dd><a href="javascript:;">选项1</a></dd>
                <dd><a href="javascript:;">选项2</a></dd>
                <dd><a href="">跳转</a></dd>
            </dl>
        </li>
        <li class="layui-nav-item">
            <a href="javascript:;">解决方案</a>
            <dl class="layui-nav-child">
                <dd><a href="">移动模块</a></dd>
                <dd><a href="">后台模版</a></dd>
                <dd><a href="">电商平台</a></dd>
            </dl>
        </li>
    </ul>
    <script src="layui/layui.all.js"></script>
    <script>
        var element = layui.element;
        var form = layui.form;
        form.render();
    </script>
</body>
</html>
```

运行结果如图 1-xx 所示。



1.9.5.2 测试属性 lay-unselect

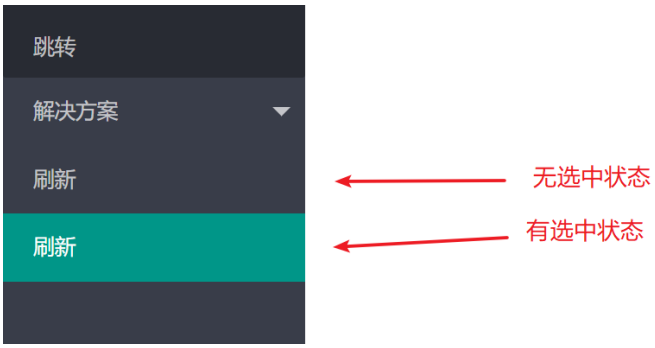
核心代码如下：

```
<li class="layui-nav-item" lay-unselect><a href="#">刷新</a></li>
<li class="layui-nav-item"><a href="#">刷新</a></li>
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <ul class="layui-nav layui-nav-tree layui-nav-side">
            <li class="layui-nav-item" lay-unselect><a href="#">刷新
</a></li>
            <li class="layui-nav-item"><a href="#">刷新</a></li>
        </ul>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            var form = layui.form;
            form.render();
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.9.6 面包屑

测试代码如下：

```
<!DOCTYPE html>
<html>
```

```
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <span class="layui-breadcrumb">
            <a href="">首页</a>
            <a href="">国际新闻</a>
            <a href="">亚太地区</a>
            <a><cite>正文</cite></a>
        </span>
        <br />
        <br />
        可以通过设置属性 lay-separator="-" 来自定义分隔符。
        <br />
        <span class="layui-breadcrumb" lay-separator="-">
            <a href="">首页</a>
            <a href="">国际新闻</a>
            <a href="">亚太地区</a>
            <a><cite>正文</cite></a>
        </span>
        <br />
        <br />
        可以作为小导航来用。
        <br />
        <span class="layui-breadcrumb" lay-separator="|">
            <a href="">娱乐</a>
            <a href="">八卦</a>
            <a href="">体育</a>
            <a href="">搞笑</a>
            <a href="">视频</a>
            <a href="">游戏</a>
            <a href="">综艺</a>
        </span>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            var form = layui.form;
            form.render();
        </script>
```

```
    </body>
</html>
```

运行结果如图 1-xx 所示。

1.10 选项卡

Tab 选项卡提供多种风格，支持响应式，支持对选项卡进行动态 CURD 操作等功能。
依赖加载组件：element。
请注意：必须加载 element 模块，相关功能才能正常使用。

1.10.1 默认 Tab 选项卡

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-tab">
            <ul class="layui-tab-title">
                <li class="layui-this">网站设置</li>
                <li>用户管理</li>
                <li>权限分配</li>
                <li>商品管理</li>
                <li>订单管理</li>
            </ul>
```

```
            <div class="layui-tab-content">
                <div class="layui-tab-item layui-show">内容1</div>
                <div class="layui-tab-item">内容2</div>
                <div class="layui-tab-item">内容3</div>
                <div class="layui-tab-item">内容4</div>
                <div class="layui-tab-item">内容5</div>
            </div>
        </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            var form = layui.form;
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。

1.10.2 Tab 简洁风格

通过对 class 追加 layui-tab-brief 类样式来设置简洁风格。

如果存在 layui-tab-item 的内容区域，在切换时自动定位到对应内容，如果不存在内容区域，则不会定位到对应内容。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
```

```
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-tab layui-tab-brief"
lay-filter="docDemoTabBrief">
            <ul class="layui-tab-title">
                <li class="layui-this">网站设置</li>
                <li>用户管理</li>
                <li>权限分配</li>
                <li>商品管理</li>
                <li>订单管理</li>
            </ul>
            <div class="layui-tab-content"></div>
        </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            var form = layui.form;
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。

1.10.3 Tab 卡片风格

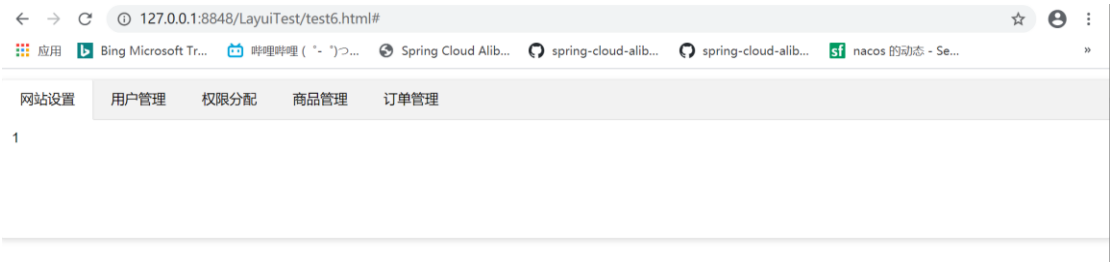
通过对 class 追加 layui-tab-card 来设置卡片风格。

测试代码如下：

```
<!DOCTYPE html>
<html>
```

```
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-tab layui-tab-card">
            <ul class="layui-tab-title">
                <li class="layui-this">网站设置</li>
                <li>用户管理</li>
                <li>权限分配</li>
                <li>商品管理</li>
                <li>订单管理</li>
            </ul>
            <div class="layui-tab-content" style="height: 100px;">
                <div class="layui-tab-item layui-show">1</div>
                <div class="layui-tab-item">2</div>
                <div class="layui-tab-item">3</div>
                <div class="layui-tab-item">4</div>
                <div class="layui-tab-item">5</div>
                <div class="layui-tab-item">6</div>
            </div>
        </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            var form = layui.form;
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



1.10.4 Tab 响应式

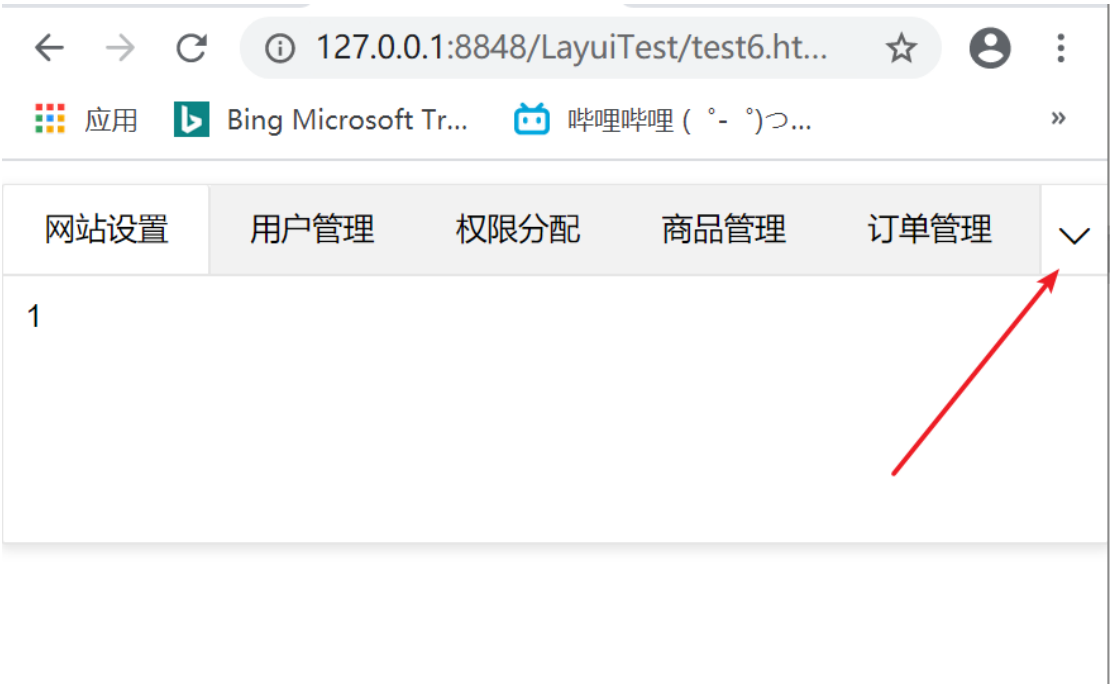
当容器的宽度不足以显示全部的选项时，即会自动出现展开图标。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-tab layui-tab-card">
            <ul class="layui-tab-title">
                <li class="layui-this">网站设置</li>
                <li>用户管理</li>
                <li>权限分配</li>
                <li>商品管理</li>
                <li>订单管理</li>
                <li>订单管理</li>
                <li>财务管理</li>
                <li>报表管理</li>
            </ul>
            <div class="layui-tab-content" style="height: 100px;">
                <div class="layui-tab-item layui-show">1</div>
                <div class="layui-tab-item">2</div>
                <div class="layui-tab-item">3</div>
                <div class="layui-tab-item">4</div>
                <div class="layui-tab-item">5</div>
                <div class="layui-tab-item">6</div>
                <div class="layui-tab-item">7</div>
                <div class="layui-tab-item">8</div>
            </div>
        </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            var form = layui.form;
            form.render();
        </script>
```

```
    </body>
</html>
```

运行结果如图 1-xx 所示。



1.10.5 带删除的 Tab

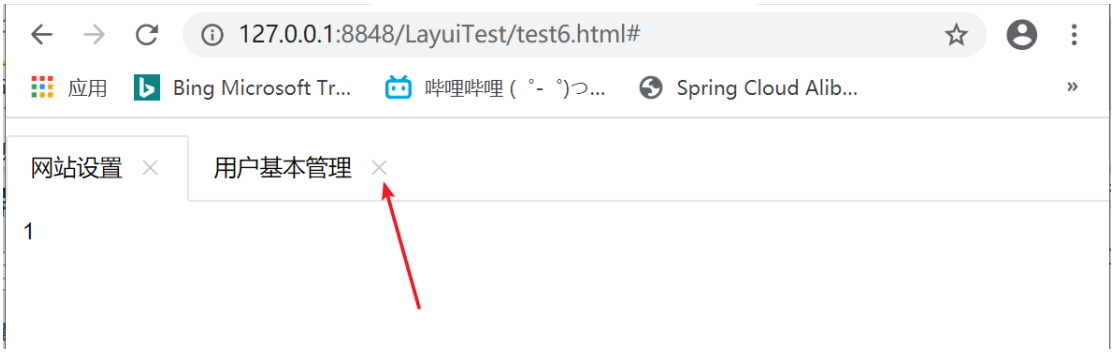
可以对父层容器设置属性 lay-allowClose="true"允许 Tab 选项卡被删除。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-tab" lay-allowClose="true">
            <ul class="layui-tab-title">
                <li class="layui-this">网站设置</li>
                <li>用户基本管理</li>
                <li>权限分配</li>
```

```
            <li>全部历史商品管理文字长一点试试</li>
            <li>订单管理</li>
        </ul>
        <div class="layui-tab-content">
            <div class="layui-tab-item layui-show">1</div>
            <div class="layui-tab-item">2</div>
            <div class="layui-tab-item">3</div>
            <div class="layui-tab-item">4</div>
            <div class="layui-tab-item">5</div>
            <div class="layui-tab-item">6</div>
        </div>
    </div>
    <script src="layui/layui.all.js"></script>
    <script>
        var element = layui.element;
        var form = layui.form;
        form.render();
    </script>
</body>
</html>
```

运行结果如图 1-xx 所示。



1.10.6 ID 焦点定位

可以通过对选项卡设置属性 lay-id="xxx"来作为唯一的匹配索引，以用于外部的定位切换。属性 lay-id 是扮演这项功能的主要角色，它是动态操作的唯一凭据。

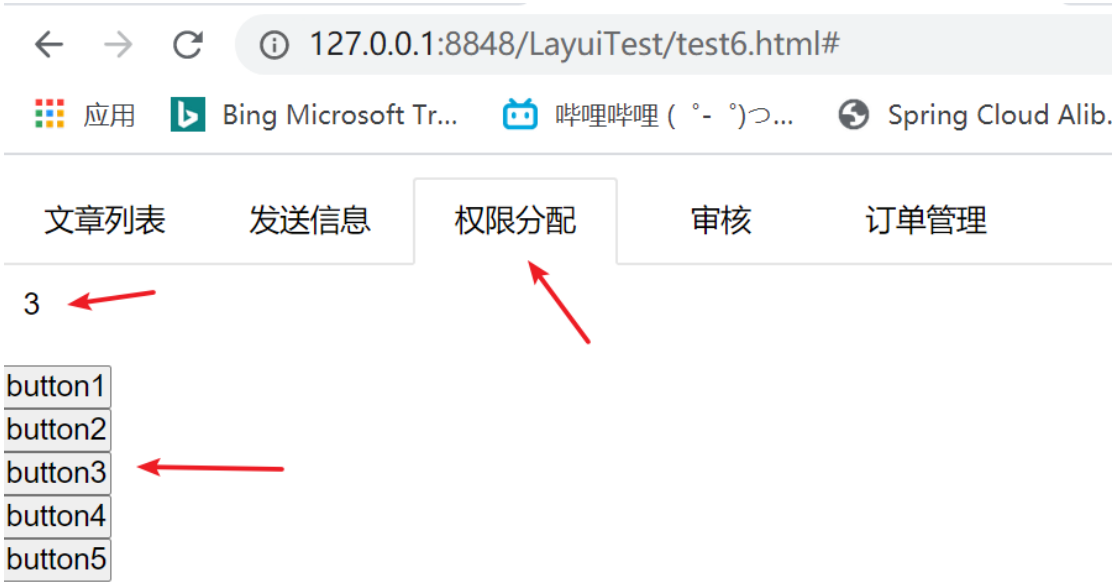
测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
```

```
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-tab" lay-filter="test1">
            <ul class="layui-tab-title">
                <li class="layui-this" lay-id="111">文章列表</li>
                <li lay-id="222">发送信息</li>
                <li lay-id="333">权限分配</li>
                <li lay-id="444">审核</li>
                <li lay-id="555">订单管理</li>
            </ul>
            <div class="layui-tab-content">
                <div class="layui-tab-item layui-show">1</div>
                <div class="layui-tab-item">2</div>
                <div class="layui-tab-item">3</div>
                <div class="layui-tab-item">4</div>
                <div class="layui-tab-item">5</div>
            </div>
        </div>
        <input type="button" value="button1"
onclick="javascript:clickMethod1()" />
        <br />
        <input type="button" value="button2"
onclick="javascript:clickMethod2()" />
        <br />
        <input type="button" value="button3"
onclick="javascript:clickMethod3()" />
        <br />
        <input type="button" value="button4"
onclick="javascript:clickMethod4()" />
        <br />
        <input type="button" value="button5"
onclick="javascript:clickMethod5()" />
        <br />
        <script>
            function clickMethod1() {
                layui.element.tabChange('test1', "111");
            }

            function clickMethod2() {
                layui.element.tabChange('test1', "222");
```

```
            }

            function clickMethod3() {
                layui.element.tabChange('test1', "333");
            }

            function clickMethod4() {
                layui.element.tabChange('test1', "444");
            }

            function clickMethod5() {
                layui.element.tabChange('test1', "555");
            }
        </script>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            var form = layui.form;
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



1.11 进度条

进度条可应用于许多业务场景，如任务完成进度、loading 进度等等，是一种较为直观的表达元素。

依赖加载组件：element。

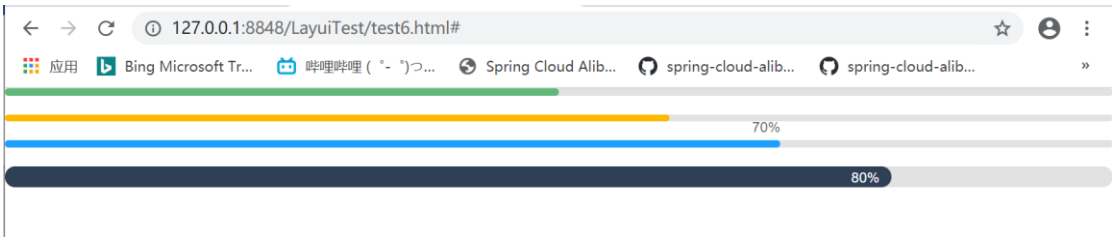
1.11.1 常规用法

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-progress">
            <div class="layui-progress-bar" lay-percent="50%"></div>
        </div>
        <br>
        <div class="layui-progress">
            <div class="layui-progress-bar layui-bg-orange" lay-percent="60%"></div>
        </div>
        <br>
        <div class="layui-progress" lay-showpercent="true">
            <div class="layui-progress-bar layui-bg-blue" lay-percent="70%"></div>
        </div>
        <br>
        <div class="layui-progress layui-progress-big" lay-showpercent="true">
            <div class="layui-progress-bar layui-bg-cyan" lay-percent="80%"></div>
        </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            var form = layui.form;
```

```
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



当对元素 class 设置 layui-progress-big 类样式时显示大尺寸进度条风格。默认风格的进度条的百分比如果开启，会在右上角显示，而大号进度条则会在内部显示。

1.11.2 显示进度比文本

通过对父级元素设置属性 lay-showPercent="yes"来开启进度比的文本显示，支持：普通数字、百分数、分数。

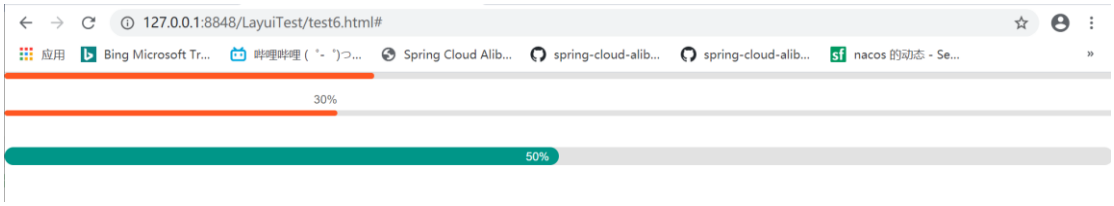
注意：默认情况下不会显示百分比文本，如果想开启，对属性 lay-showPercent 设置任意值即可，例如：yes。但如果不想显示，千万别设置 no 或者 false，直接剔除该属性即可。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-progress" lay-showPercent="true">
            <div class="layui-progress-bar layui-bg-red"
lay-percent="1/3"></div>
        </div>
        <br />
        <br />
        <div class="layui-progress" lay-showPercent="yes">
            <div class="layui-progress-bar layui-bg-red"
lay-percent="30%"></div>
```

```
        </div>
        <br />
        <br />
        <div class="layui-progress layui-progress-big"
lay-showPercent="yes">
            <div class="layui-progress-bar layui-bg-green"
lay-percent="50%"></div>
        </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            var form = layui.form;
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



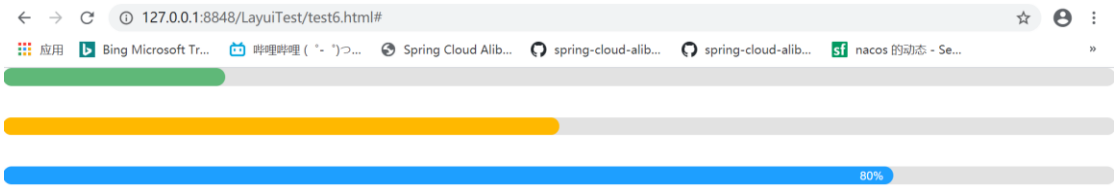
1.11.3 大号进度条

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-progress layui-progress-big">
            <div class="layui-progress-bar" lay-percent="20%"></div>
        </div>
        <br />
        <br />
```

```
        <div class="layui-progress layui-progress-big">
            <div class="layui-progress-bar layui-bg-orange"
lay-percent="50%"></div>
        </div>
        <br />
        <br />
        <div class="layui-progress layui-progress-big"
lay-showPercent="true">
            <div class="layui-progress-bar layui-bg-blue"
lay-percent="80%"></div>
        </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            var form = layui.form;
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



1.12 面板

面板通常是指一个独立的容器，而折叠面板则能有效地节省页面的可视面积，非常适合应用于 QA 说明、帮助文档等等。

依赖加载组件：element。

1.12.1 卡片面板

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
```

```
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div style="padding: 5px;background-color: #cccccc;"
class="layui-row layui-col-space5">
            <div class="layui-col-md6">
                <div class="layui-card">
                    <div class="layui-card-header">卡片面板</div>
                    <div class="layui-card-body">
                        卡片式面板面板通常用于非白色背景色的主体内<br>
                        从而映衬出边框投影
                    </div>
                </div>
            </div>
            <div class="layui-col-md6">
                <div class="layui-card">
                    <div class="layui-card-header">卡片面板</div>
                    <div class="layui-card-body">
                        卡片式面板面板通常用于非白色背景色的主体内<br>
                        从而映衬出边框投影
                    </div>
                </div>
            </div>
        </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            var form = layui.form;
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。

如果网页采用白色做为背景，不建议使用卡片面板。

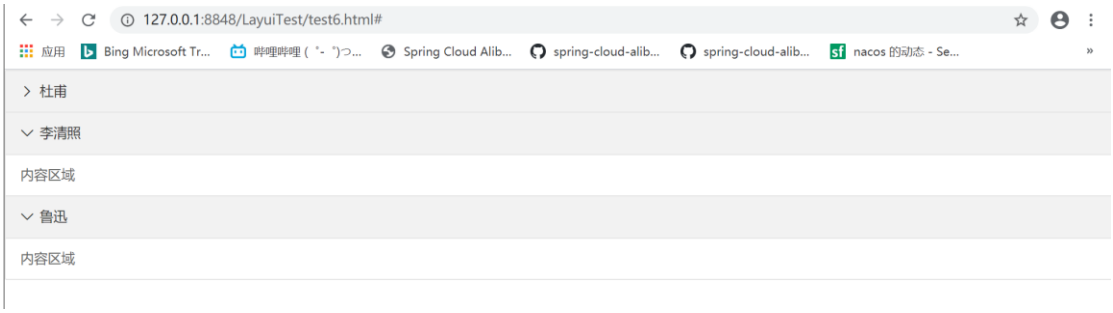
1.12.2 折叠面板

通过对内容元素设置 class 为 layui-show 来选择性初始展开某一个面板，会自动呈现状态图标。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-collapse">
            <div class="layui-colla-item">
                <h2 class="layui-colla-title">杜甫</h2>
                <div class="layui-colla-content layui-show">内容区域1</div>
            </div>
            <div class="layui-colla-item">
                <h2 class="layui-colla-title">李清照</h2>
                <div class="layui-colla-content layui-show">内容区域2</div>
            </div>
            <div class="layui-colla-item">
                <h2 class="layui-colla-title">鲁迅</h2>
                <div class="layui-colla-content">内容区域3</div>
            </div>
        </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            var form = layui.form;
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



1.12.3 开启手风琴

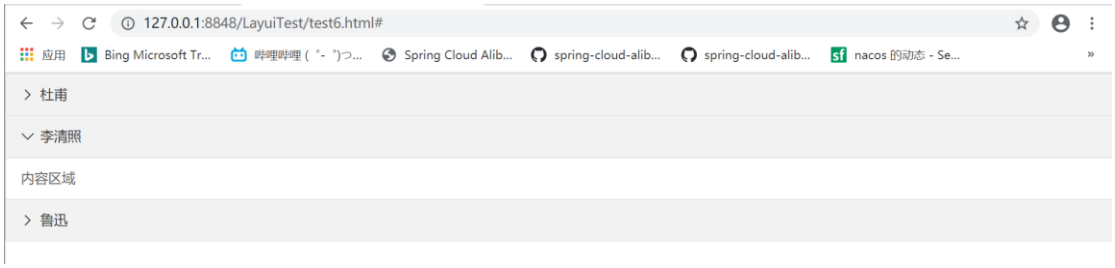
在折叠面板的父容器设置属性 lay-accordion 来开启手风琴风格，在进行折叠操作时，始终只会展现当前选中的面板，其它面板呈隐藏的状态。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-collapse" lay-accordion>
            <div class="layui-colla-item">
                <h2 class="layui-colla-title">杜甫</h2>
                <div class="layui-colla-content layui-show">内容区域</div>
            </div>
            <div class="layui-colla-item">
                <h2 class="layui-colla-title">李清照</h2>
                <div class="layui-colla-content layui-show">内容区域</div>
            </div>
            <div class="layui-colla-item">
                <h2 class="layui-colla-title">鲁迅</h2>
                <div class="layui-colla-content layui-show">内容区域</div>
            </div>
        </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
```

```
            var form = layui.form;
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



1.13 表格

可以通过内置的自定义属性对 Table 表格进行风格的多样化设置。

1.13.1 常规用法

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <table class="layui-table">
            <colgroup>
                <col width="150">
                <col width="200">
                <col>
            </colgroup>
            <thead>
                <tr>
                    <th>昵称</th>
```

```
                <th>加入时间</th>
                <th>签名</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>贤心</td>
                <td>2016-11-29</td>
                <td>人生就像是一场修行</td>
            </tr>
            <tr>
                <td>许闲心</td>
                <td>2016-11-28</td>
                <td>于千万人之中遇见你所遇见的人，于千万年之中，时间的无涯的荒野里…</td>
            </tr>
        </tbody>
    </table>
    <script src="layui/layui.all.js"></script>
    <script>
        var table = layui.table;
        var form = layui.form;
        form.render();
    </script>
</body>
</html>
```

运行结果如图 1-xx 所示。

1.13.2 基础属性与用法

表格 Table 基础属性与用法如图 1-xx 所示。

静态表格支持以下基础属性，可定义不同风格/尺寸的表格样式：

属性名	属性值	备注
lay-even	无	用于开启 *隔行* 背景，可与其它属性一起使用
lay-skin="属性值"	line （行边框风格） row （列边框风格） nob （无边框风格）	若使用默认风格不设置该属性即可
lay-size="属性值"	sm （小尺寸） lg （大尺寸）	若使用默认尺寸不设置该属性即可

将你所需要的基础属性写在table标签上即可，如（一个带有隔行背景，且行边框风格的大尺寸表格）：

```HTML
<table lay-even lay-skin="line" lay-size="lg">
…
</table>

```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        默认风格：
        <br />
        <table class="layui-table">
            <colgroup>
                <col width="150">
                <col width="200">
                <col>
            </colgroup>
            <thead>
                <tr>
                    <th>昵称</th>
                    <th>加入时间</th>
                    <th>签名</th>
```

```
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>贤心</td>
                <td>2016-11-29</td>
                <td>人生就像是一场修行</td>
            </tr>
            <tr>
                <td>许闲心</td>
                <td>2016-11-28</td>
                <td>于千万人之中遇见你所遇见的人，于千万年之中，时间的无涯
的荒野里…</td>
            </tr>
        </tbody>
    </table>
    <br />
    <br />
    隔行变色：
    <br />
    <table class="layui-table" lay-even>
        <colgroup>
            <col width="150">
            <col width="200">
            <col>
        </colgroup>
        <thead>
            <tr>
                <th>昵称</th>
                <th>加入时间</th>
                <th>签名</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>贤心</td>
                <td>2016-11-29</td>
                <td>人生就像是一场修行</td>
            </tr>
            <tr>
                <td>许闲心</td>
                <td>2016-11-28</td>
                <td>于千万人之中遇见你所遇见的人，于千万年之中，时间的无涯
的荒野里…</td>
```

```
            </tr>
        </tbody>
    </table>
    <br />
    <br />
    隔行变色+只有横线:
    <br />
    <table class="layui-table" lay-even lay-skin="line">
        <colgroup>
            <col width="150">
            <col width="200">
            <col>
        </colgroup>
        <thead>
            <tr>
                <th>昵称</th>
                <th>加入时间</th>
                <th>签名</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>贤心</td>
                <td>2016-11-29</td>
                <td>人生就像是一场修行</td>
            </tr>
            <tr>
                <td>许闲心</td>
                <td>2016-11-28</td>
                <td>于千万人之中遇见你所遇见的人，于千万年之中，时间的无涯的荒野里…</td>
            </tr>
        </tbody>
    </table>
    <br />
    <br />
    隔行变色+只有竖线:
    <br />
    <table class="layui-table" lay-even lay-skin="row">
        <colgroup>
            <col width="150">
            <col width="200">
            <col>
        </colgroup>
```

```
            <thead>
                <tr>
                    <th>昵称</th>
                    <th>加入时间</th>
                    <th>签名</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>贤心</td>
                    <td>2016-11-29</td>
                    <td>人生就像是一场修行</td>
                </tr>
                <tr>
                    <td>许闲心</td>
                    <td>2016-11-28</td>
                    <td>于千万人之中遇见你所遇见的人，于千万年之中，时间的无涯的荒野里…</td>
                </tr>
            </tbody>
        </table>
        <br />
        <br />
        隔行变色+没有线：
        <br />
        <table class="layui-table" lay-even lay-skin="nob">
            <colgroup>
                <col width="150">
                <col width="200">
                <col>
            </colgroup>
            <thead>
                <tr>
                    <th>昵称</th>
                    <th>加入时间</th>
                    <th>签名</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>贤心</td>
                    <td>2016-11-29</td>
                    <td>人生就像是一场修行</td>
                </tr>
```

```
                <tr>
                    <td>许闲心</td>
                    <td>2016-11-28</td>
                    <td>于千万人之中遇见你所遇见的人，于千万年之中，时间的无涯的荒野里…</td>
                </tr>
            </tbody>
        </table>
        <br />
        <br />
        小表格:
        <br />
        <table class="layui-table" lay-size="sm">
            <colgroup>
                <col width="150">
                <col width="200">
                <col>
            </colgroup>
            <thead>
                <tr>
                    <th>昵称</th>
                    <th>加入时间</th>
                    <th>签名</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>贤心</td>
                    <td>2016-11-29</td>
                    <td>人生就像是一场修行</td>
                </tr>
                <tr>
                    <td>许闲心</td>
                    <td>2016-11-28</td>
                    <td>于千万人之中遇见你所遇见的人，于千万年之中，时间的无涯的荒野里…</td>
                </tr>
            </tbody>
        </table>
        <br />
        <br />
        大表格:
        <br />
        <table class="layui-table" lay-size="lg">
```

```
            <colgroup>
                <col width="150">
                <col width="200">
                <col>
            </colgroup>
            <thead>
                <tr>
                    <th>昵称</th>
                    <th>加入时间</th>
                    <th>签名</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>贤心</td>
                    <td>2016-11-29</td>
                    <td>人生就像是一场修行</td>
                </tr>
                <tr>
                    <td>许闲心</td>
                    <td>2016-11-28</td>
                    <td>于千万人之中遇见你所遇见的人，于千万年之中，时间的无涯的荒野里…</td>
                </tr>
            </tbody>
        </table>
        <script src="layui/layui.all.js"></script>
        <script>
            var table = layui.table;
            var form = layui.form;
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。

1.14 徽章

徽章是一个修饰性的元素，它们本身细小而并不显眼，但掺杂在其它元素中就显得尤为突出了。页面往往因徽章的陪衬，而显得十分和谐和精致。

1.14.1 快速使用

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        小圆点，通过 layui-badge-dot 来定义，里面不能加文字
        <br />
        <span class="layui-badge-dot"></span>
        <span class="layui-badge-dot layui-bg-orange"></span>
        <span class="layui-badge-dot layui-bg-green"></span>
        <span class="layui-badge-dot layui-bg-cyan"></span>
        <span class="layui-badge-dot layui-bg-blue"></span>
        <span class="layui-badge-dot layui-bg-black"></span>
        <span class="layui-badge-dot layui-bg-gray"></span>
        <br />
        <br />
        椭圆体，通过 layui-badge 来定义。事实上我们把这个视作为主要使用方式
        <br />
        <span class="layui-badge">6</span>
        <span class="layui-badge">99</span>
        <span class="layui-badge">61728</span>
        <span class="layui-badge">赤</span>
        <span class="layui-badge layui-bg-orange">橙</span>
        <span class="layui-badge layui-bg-green">绿</span>
        <span class="layui-badge layui-bg-cyan">青</span>
        <span class="layui-badge layui-bg-blue">蓝</span>
        <span class="layui-badge layui-bg-black">黑</span>
        <span class="layui-badge layui-bg-gray">灰</span>
        <br />
        <br />
        边框体，通过 layui-badge-rim 来定义
        <br />
        <span class="layui-badge-rim">6</span>
        <span class="layui-badge-rim">Hot</span>
        <script src="layui/layui.all.js"></script>
```

```
        <script>
            var form = layui.form;
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。

徽章具有三种不同的风格类型：小圆点、椭圆体、边框体。

1.14.2 与其它元素的搭配

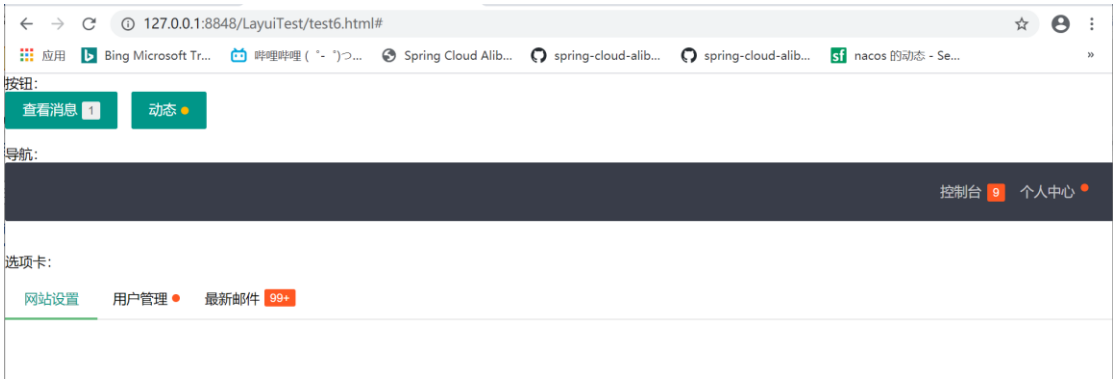
徽章主要是修饰作用，因此必不可少要与几乎所有的元素搭配，Layui 目前对按钮，导航，选项卡这 3 个元素内置了徽章的排版支持。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        按钮：
        <br />
        <button class="layui-btn">查看消息<span class="layui-badge
layui-bg-gray">1</span></button>
```

```
        <button class="layui-btn">动态<span class="layui-badge-dot
layui-bg-orange"></span></button>
        <br />
        <br />
        导航：
        <br />
        <ul class="layui-nav" style="text-align: right;">
            <-- 小Tips：这里有没有发现，设置导航靠右对齐（或居中对齐）其实非常
简单 -->
                <li class="layui-nav-item">
                    <a href="">控制台<span class="layui-badge">9</span></a>
                </li>
                <li class="layui-nav-item">
                    <a href="">个人中心<span
class="layui-badge-dot"></span></a>
                </li>
        </ul>
        <br />
        <br />
        选项卡：
        <br />
        <div class="layui-tab layui-tab-brief">
            <ul class="layui-tab-title">
                <li class="layui-this">网站设置</li>
                <li>用户管理<span class="layui-badge-dot"></span></li>
                <li>最新邮件<span class="layui-badge">99+</span></li>
            </ul>
            <div class="layui-tab-content"></div>
        </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var form = layui.form;
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



1.15 时间线

将时间抽象到二维平面，垂直呈现一段从过去到现在的故事。

示例代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <ul class="layui-timeline">
            <li class="layui-timeline-item">
                <i class="layui-icon layui-timeline-axis">&#xe63f;</i>
                <div class="layui-timeline-content layui-text">
                    <h3 class="layui-timeline-title">8月18日</h3>
                    <p>
                        layui 2.0 的一切准备工作似乎都已到位。发布之弦，一触即
发。
                        <br>不枉近百个日日夜夜与之为伴。因小而大，因弱而强。
                        <br>无论它能走多远，抑或如何支撑？至少我曾倾注全心，无
怒无悔 <i class="layui-icon"> </i>
                    </p>
                </div>
            </li>
            <li class="layui-timeline-item">
                <i class="layui-icon layui-timeline-axis">&#xe63f;</i>
                <div class="layui-timeline-content layui-text">
```

```
                    <h3 class="layui-timeline-title">8月16日</h3>
                    <p>杜甫的思想核心是儒家的仁政思想，他有“<em>致君尧舜上，再使风俗淳</em>”的宏伟抱负。个人最爱的名篇有：</p>
                    <ul>
                        <li>《登高》</li>
                        <li>《茅屋为秋风所破歌》</li>
                    </ul>
                </div>
            </li>
            <li class="layui-timeline-item">
                <i class="layui-icon layui-timeline-axis">&#xe63f;</i>
                <div class="layui-timeline-content layui-text">
                    <h3 class="layui-timeline-title">8月15日</h3>
                    <p>
                        中国人民抗日战争胜利72周年
                        <br>常常在想，尽管对这个国家有这样那样的抱怨，但我们的确生在了最好的时代
                        <br>铭记、感恩
                        <br>所有为中华民族浴血奋战的英雄将士
                        <br>永垂不朽
                    </p>
                </div>
            </li>
            <li class="layui-timeline-item">
                <i class="layui-icon layui-timeline-axis">&#xe63f;</i>
                <div class="layui-timeline-content layui-text">
                    <div class="layui-timeline-title">过去</div>
                </div>
            </li>
        </ul>
        <script src="layui/layui.all.js"></script>
        <script>
            var form = layui.form;
            form.render();
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。

时间线中的内容区域可自由填充。

1.16 辅助

本篇主要集中罗列页面中的一些简单辅助元素，如：引用块、字段集区块、横线等等，这些元素都无需依赖任何模块。

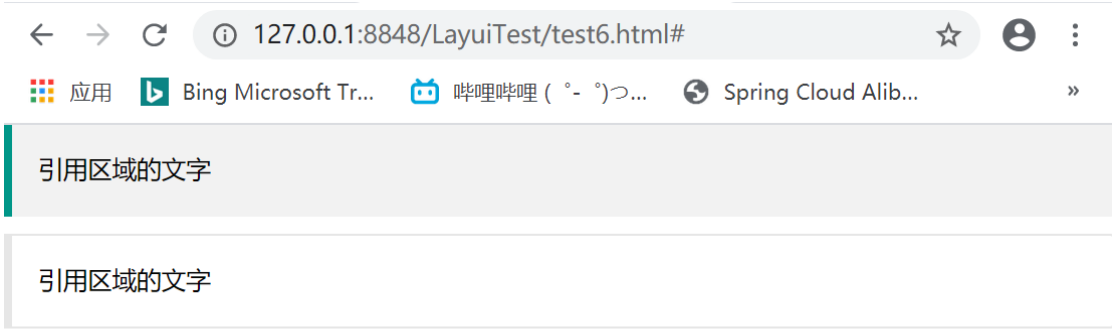
1.16.1 引用区块

示例代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
```

```
        <blockquote class="layui-elem-quote">引用区域的文字</blockquote>
        <blockquote class="layui-elem-quote layui-quote-nm">引用区域的文字
</blockquote>
        <script src="layui/layui.all.js"></script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



目前内置了上述两种风格。

1.16.2 字段集区块

示例代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <fieldset class="layui-elem-field">
            <legend>字段集区块 - 默认风格</legend>
            <div class="layui-field-box">
                内容区域
            </div>
        </fieldset>

        <fieldset class="layui-elem-field layui-field-title">
            <legend>字段集区块 - 横线风格</legend>
```

```
            <div class="layui-field-box">
                内容区域
            </div>
        </fieldset>
        你可以把它看作是一个有标题的横线
        <script src="layui/layui.all.js"></script>
    </body>
</html>
```

运行结果如图 1-xx 所示。

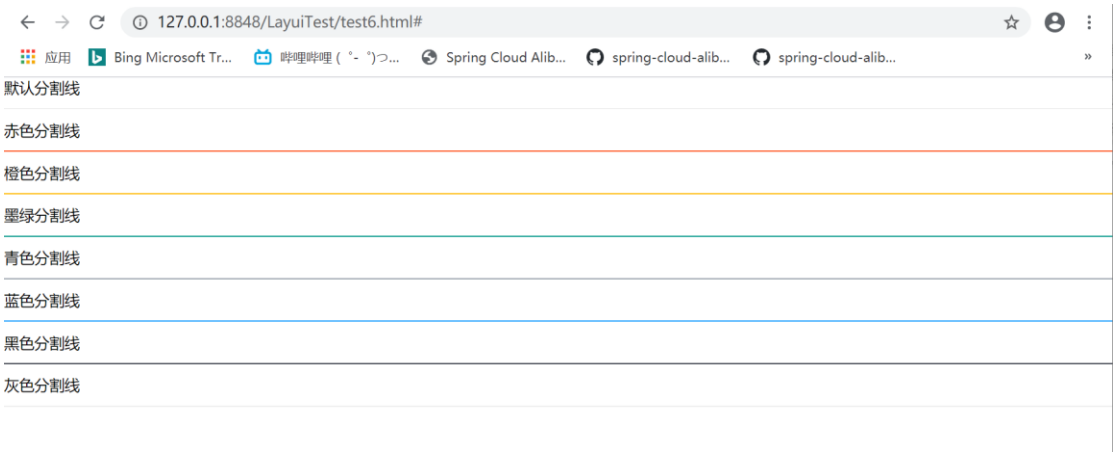
1.16.3 横线

示例代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        默认分割线
        <hr>
        赤色分割线
        <hr class="layui-bg-red">
        橙色分割线
```

```
        <hr class="layui-bg-orange">
        墨绿分割线
        <hr class="layui-bg-green">
        青色分割线
        <hr class="layui-bg-cyan">
        蓝色分割线
        <hr class="layui-bg-blue">
        黑色分割线
        <hr class="layui-bg-black">
        灰色分割线
        <hr class="layui-bg-gray">
        <script src="layui/layui.all.js"></script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



1.17 后台布局整合表格

参考代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
    </head>
    <body class="layui-layout-body">
        <div class="layui-layout layui-layout-admin">
            <div class="layui-header">
```

```
            <div class="layui-logo">layui 后台布局</div>
            <!-- 头部区域（可配合layui已有的水平导航） -->
            <ul class="layui-nav layui-layout-left">
                <li class="layui-nav-item"><a href="">控制台</a></li>
                <li class="layui-nav-item"><a href="">商品管理</a></li>
                <li class="layui-nav-item"><a href="">用户</a></li>
                <li class="layui-nav-item">
                    <a href="javascript:;">其它系统</a>
                    <dl class="layui-nav-child">
                        <dd><a href="">邮件管理</a></dd>
                        <dd><a href="">消息管理</a></dd>
                        <dd><a href="">授权管理</a></dd>
                    </dl>
                </li>
            </ul>
            <ul class="layui-nav layui-layout-right">
                <li class="layui-nav-item">
                    <a href="javascript:;">
                        <img src="http://t.cn/RCzsdCq"
class="layui-nav-img">
                        贤心
                    </a>
                    <dl class="layui-nav-child">
                        <dd><a href="">基本资料</a></dd>
                        <dd><a href="">安全设置</a></dd>
                    </dl>
                </li>
                <li class="layui-nav-item"><a href="">退了</a></li>
            </ul>
        </div>

        <div class="layui-side layui-bg-black">
            <div class="layui-side-scroll">
                <!-- 左侧导航区域（可配合layui已有的垂直导航） -->
                <ul class="layui-nav layui-nav-tree" lay-filter="test">
                    <li class="layui-nav-item layui-nav-itemed">
                        <a class="" href="javascript:;">用户管理</a>
                        <dl class="layui-nav-child">
                            <dd><a href="javascript:;">增加用户</a></dd>
                            <dd><a href="javascript:;">查询用户</a></dd>
                            <dd><a href="javascript:;">注销用户</a></dd>
                            <dd><a href="javascript:;">用户统计</a></dd>
                        </dl>
                    </li>
```

```
                    <li class="layui-nav-item layui-nav-itemed">
                        <a class="" href="javascript:;">用户管理</a>
                        <dl class="layui-nav-child">
                            <dd><a href="javascript:;">增加用户</a></dd>
                            <dd><a href="javascript:;">查询用户</a></dd>
                            <dd><a href="javascript:;">注销用户</a></dd>
                            <dd><a href="javascript:;">用户统计</a></dd>
                        </dl>
                    </li>
                    <li class="layui-nav-item layui-nav-itemed">
                        <a class="" href="javascript:;">用户管理</a>
                        <dl class="layui-nav-child">
                            <dd><a href="javascript:;">增加用户</a></dd>
                            <dd><a href="javascript:;">查询用户</a></dd>
                            <dd><a href="javascript:;">注销用户</a></dd>
                            <dd><a href="javascript:;">用户统计</a></dd>
                        </dl>
                    </li>
                    <li class="layui-nav-item">
                        <a href="javascript:;">解决方案</a>
                        <dl class="layui-nav-child">
                            <dd><a href="javascript:;">列表一</a></dd>
                            <dd><a href="javascript:;">列表二</a></dd>
                            <dd><a href="">超链接</a></dd>
                        </dl>
                    </li>
                    <li class="layui-nav-item">
                        <a href="javascript:;">云市场</a>
                        <dl class="layui-nav-child">
                            <dd><a href="javascript:;">增加用户</a></dd>
                            <dd><a href="javascript:;">查询用户</a></dd>
                            <dd><a href="javascript:;">注销用户</a></dd>
                            <dd><a href="javascript:;">用户统计</a></dd>
                        </dl>
                    </li>
                    <li class="layui-nav-item"><a href="">发布商品
</a></li>
                </ul>
            </div>
        </div>

        <div class="layui-body">
            <div style="padding: 5px 15px;">
                <div class="layui-row layui-col-space10">
```

```
                    <div class="layui-col-md12">
                        <div>
                            <table class="layui-table" lay-even>
                                <colgroup>
                                    <col width="100">
                                    <col>
                                    <col width="100">
                                    <col>
                                    <col width="100">
                                    <col>
                                    <col width="100">
                                    <col>
                                </colgroup>
                                <tbody>
                                    <tr>
                                        <td>账号</td>
                                        <td><input type="text" id="xxxxx"
placeholder="请输入账号" autocomplete="off" class="layui-input"></td>
                                        <td>密码</td>
                                        <td><input type="text" id="xxxxx"
placeholder="请输入账号" autocomplete="off" class="layui-input"></td>
                                        <td>年龄</td>
                                        <td><input type="text" id="xxxxx"
placeholder="请输入账号" autocomplete="off" class="layui-input"></td>
                                        <td>手机号</td>
                                        <td><input type="text" id="xxxxx"
placeholder="请输入账号" autocomplete="off" class="layui-input"></td>
                                    </tr>
                                    <tr>
                                        <td>邮箱</td>
                                        <td><input type="text" id="xxxxx"
placeholder="请输入账号" autocomplete="off" class="layui-input"></td>
                                        <td>QQ号</td>
                                        <td><input type="text" id="xxxxx"
placeholder="请输入账号" autocomplete="off" class="layui-input"></td>
                                        <td>身份证号</td>
                                        <td><input type="text" id="xxxxx"
placeholder="请输入账号" autocomplete="off" class="layui-input"></td>
                                        <td>性别</td>
                                        <td>
                                            <div class="layui-form">
                                                <select name="city"
lay-verify="">
                                                    <option value="">请选
```

```
择一个城市</option>
                                                    <option value="010">
北京</option>
                                                    <option value="021">
上海</option>
                                                    <option value="0571">
杭州</option>
                                                </select>
                                            </div>
                                        </td>
                                    </tr>
                                    <tr>
                                        <td colspan="8" style="text-align:
center;">
                                            <button type="button"
class="layui-btn layui-btn-normal" style="width: 100px;">查询</button>
                                            <button type="button"
class="layui-btn layui-btn-warm" style="width: 100px;">重置</button>
                                        </td>
                                    </tr>
                                </tbody>
                            </table>
                        </div>
                    </div>
                </div>
                <div class="layui-row layui-col-space10">
                    <div class="layui-col-md12">
                        <div>
                            <table class="layui-table" lay-even>
                                <colgroup>
                                    <col width="50">
                                    <col>
                                    <col>
                                    <col width="80">
                                    <col width="150">
                                    <col>
                                    <col width="180">
                                    <col width="200">
                                    <col>
                                </colgroup>
                                <thead>
                                    <tr>
                                        <th>ID</th>
                                        <th>账号</th>
```

```
                <th>密码</th>
                <th>年龄</th>
                <th>手机号</th>
                <th>邮箱</th>
                <th>QQ号</th>
                <th>身份证号</th>
                <th>性别</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>1</td>
                <td>高洪岩</td>
                <td>123456789</td>
                <td>18</td>
                <td>13812345678</td>
                <td>13812345678@qq.com</td>
                <td>123456789</td>
                <td>210181199001011234</td>
                <td>男</td>
            </tr>
            <tr>
                <td>1</td>
                <td>高洪岩</td>
                <td>123456789</td>
                <td>18</td>
                <td>13812345678</td>
                <td>13812345678@qq.com</td>
                <td>123456789</td>
                <td>210181199001011234</td>
                <td>男</td>
            </tr>
            <tr>
                <td>1</td>
                <td>高洪岩</td>
                <td>123456789</td>
                <td>18</td>
                <td>13812345678</td>
                <td>13812345678@qq.com</td>
                <td>123456789</td>
                <td>210181199001011234</td>
                <td>男</td>
            </tr>
            <tr>
```

```
                <td>1</td>
                <td>高洪岩</td>
                <td>123456789</td>
                <td>18</td>
                <td>13812345678</td>
                <td>13812345678@qq.com</td>
                <td>123456789</td>
                <td>210181199001011234</td>
                <td>男</td>
            </tr>
            <tr>
                <td>1</td>
                <td>高洪岩</td>
                <td>123456789</td>
                <td>18</td>
                <td>13812345678</td>
                <td>13812345678@qq.com</td>
                <td>123456789</td>
                <td>210181199001011234</td>
                <td>男</td>
            </tr>
            <tr>
                <td>1</td>
                <td>高洪岩</td>
                <td>123456789</td>
                <td>18</td>
                <td>13812345678</td>
                <td>13812345678@qq.com</td>
                <td>123456789</td>
                <td>210181199001011234</td>
                <td>男</td>
            </tr>
            <tr>
                <td>1</td>
                <td>高洪岩</td>
                <td>123456789</td>
                <td>18</td>
                <td>13812345678</td>
                <td>13812345678@qq.com</td>
                <td>123456789</td>
                <td>210181199001011234</td>
                <td>男</td>
            </tr>
            <tr>
```

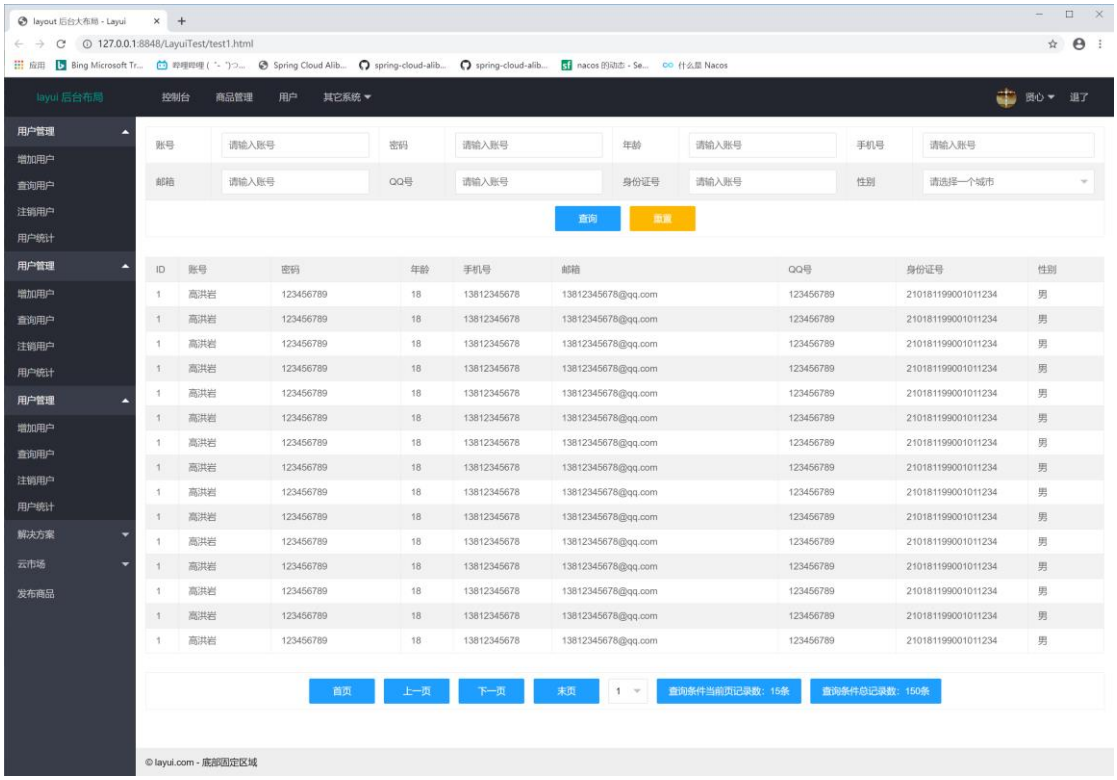
```
    <td>1</td>
    <td>高洪岩</td>
    <td>123456789</td>
    <td>18</td>
    <td>13812345678</td>
    <td>13812345678@qq.com</td>
    <td>123456789</td>
    <td>210181199001011234</td>
    <td>男</td>
</tr>
<tr>
    <td>1</td>
    <td>高洪岩</td>
    <td>123456789</td>
    <td>18</td>
    <td>13812345678</td>
    <td>13812345678@qq.com</td>
    <td>123456789</td>
    <td>210181199001011234</td>
    <td>男</td>
</tr>
<tr>
    <td>1</td>
    <td>高洪岩</td>
    <td>123456789</td>
    <td>18</td>
    <td>13812345678</td>
    <td>13812345678@qq.com</td>
    <td>123456789</td>
    <td>210181199001011234</td>
    <td>男</td>
</tr>
<tr>
    <td>1</td>
    <td>高洪岩</td>
    <td>123456789</td>
    <td>18</td>
    <td>13812345678</td>
    <td>13812345678@qq.com</td>
    <td>123456789</td>
    <td>210181199001011234</td>
    <td>男</td>
</tr>
<tr>
```

```
        <td>1</td>
        <td>高洪岩</td>
        <td>123456789</td>
        <td>18</td>
        <td>13812345678</td>
        <td>13812345678@qq.com</td>
        <td>123456789</td>
        <td>210181199001011234</td>
        <td>男</td>
    </tr>
    <tr>
        <td>1</td>
        <td>高洪岩</td>
        <td>123456789</td>
        <td>18</td>
        <td>13812345678</td>
        <td>13812345678@qq.com</td>
        <td>123456789</td>
        <td>210181199001011234</td>
        <td>男</td>
    </tr>
    <tr>
        <td>1</td>
        <td>高洪岩</td>
        <td>123456789</td>
        <td>18</td>
        <td>13812345678</td>
        <td>13812345678@qq.com</td>
        <td>123456789</td>
        <td>210181199001011234</td>
        <td>男</td>
    </tr>
    <tr>
        <td>1</td>
        <td>高洪岩</td>
        <td>123456789</td>
        <td>18</td>
        <td>13812345678</td>
        <td>13812345678@qq.com</td>
        <td>123456789</td>
        <td>210181199001011234</td>
        <td>男</td>
    </tr>
</tbody>
```

```
                            </table>

                        </div>
                    </div>
                </div>
                <div class="layui-row layui-col-space10">
                    <div class="layui-col-md12">
                        <div>
                            <table class="layui-table">
                                <tr>
                                    <td colspan="8" style="text-align:
center;">
                                        <button type="button"
class="layui-btn layui-btn-normal" style="width:
100px;margin-bottom:0px;">首页</button>
                                        <button type="button"
class="layui-btn layui-btn-normal" style="width:
100px;margin-bottom:0px;">上一页</button>
                                        <button type="button"
class="layui-btn layui-btn-normal" style="width:
100px;margin-bottom:0px;">下一页</button>
                                        <button type="button"
class="layui-btn layui-btn-normal" style="width:
100px;margin-bottom:0px;">末页</button>
                                        <div class="layui-form
layui-input-inline" style="color: #000000; width: 60px; margin-left:
10px;">
                                            <select name="gotoPage">
                                                <option value="1"
selected="selected">1</option>
                                                <option
value="2">2</option>
                                                <option
value="3">3</option>
                                                <option
value="4">4</option>
                                                <option
value="5">5</option>
                                                <option
value="6">6</option>
                                                <option
value="7">7</option>
                                                <option
value="8">8</option>
```

```
                                                <option
value="9">9</option>
                                            </select>
                                        </div>
                                        <button type="button"
class="layui-btn layui-btn-normal" style="margin-bottom:0px;margin-left:
10px;">查询条件当前页记录数：15条</button>
                                        <button type="button"
class="layui-btn layui-btn-normal" style="margin-bottom:0px;">查询条件总
记录数：150条</button>
                                    </td>
                                </tr>
                                </tbody>
                            </table>
                        </div>
                    </div>
                </div>
            </div>
            <div class="layui-footer">
                <!-- 底部固定区域 -->
                © layui.com - 底部固定区域
            </div>
        </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            var form = layui.form;
            form.render();
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.18 弹层组件 layer

layer 是一款近年来备受青睐的 Web 弹层组件，具备全方位的解决方案，致力于服务各水平段的开发人员，会非常轻松地拥有丰富友好的操作体验。在与同类组件的比较中，layer 总是能轻易获胜。她尽可能地在以更少的代码展现更强劲的功能，且格外注重性能的提升、易用和实用性，正因如此，越来越多的开发者将关注点投在了 layer（已被 11487543 人次关注）。layer 甚至兼容了包括 IE6 在内的所有主流浏览器。提供数量可观的接口，使得可以自定义太多需要的风格，每一种弹层模式各具特色，广受欢迎。layer 采用 MIT 开源许可证，将会永久性提供无偿服务。

layer 至今仍作为 Layui 的代表作，受众广泛并非偶然，这正是数年来开发者的坚持、不弃的信念，不断的完善和维护、不断建设和提升社区服务，在 Web 开发者的圈子里口口相传，乃至成为今天的 Layui 最强劲的源动力。目前，layer 已经成为国内最多人使用的 Web 弹层组件，GitHub 成绩为 Stars 5000+，官网累计下载量达 50w+，大概有 30 万不同规模的 Web 平台使用过 layer。再次强调，layer 只是作为 Layui 的一个弹层模块，由于其用户基数较大，所以至今仍把它作为独立组件来维护。

模块加载名称：layer。

layer 独立版本的官网地址为：

http://layer.layui.com

1.18.1 使用方式

layer 有两种使用方式，如图 1-xx 所示。

由于layer可以独立使用，也可以通过Layui模块化使用。所以请按照你的实际需求来选择。

场景	用前准备	调用方式
1. 作为独立组件使用	如果你只是单独想使用 layer，你可以去 layer 独立版本官网下载组件包。你需要在你的页面引入 *jQuery* 1.8以上的任意版本，并引入 *layer.js* 。	通过script标签引入layer.js后，直接用即可。 参考：快速上手
2. layui 模块化使用	如果你使用的是 layui，那么你直接在官网下载 layui 框架即可，无需引入 jQuery 和 layer.js，但需要引入 *layui.css* 和 *layui.js*	通过 *layui.use('layer', callback)* 加载模块

1.18.1.1 独立使用

当独立使用时，需要到官网：

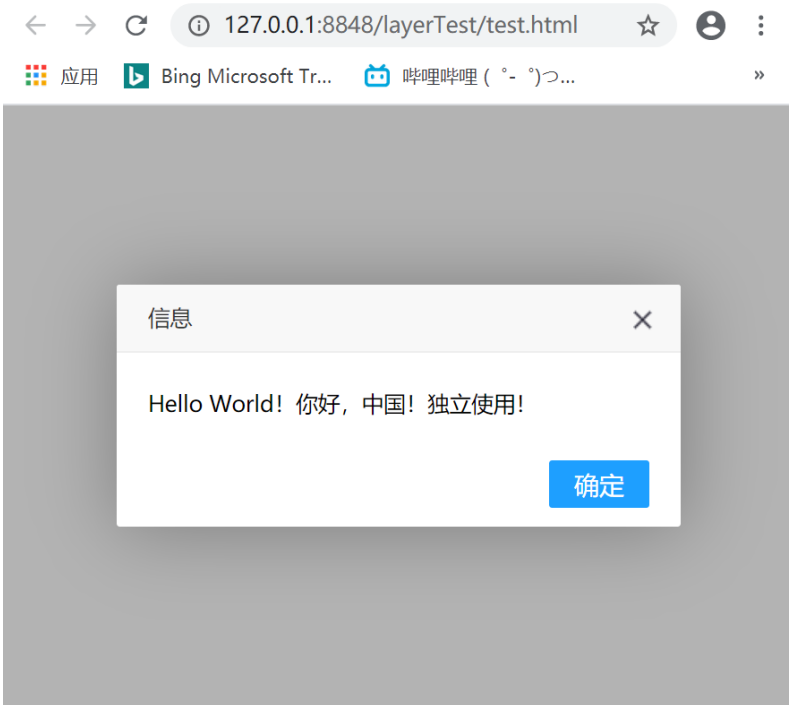
https://layer.layui.com/

下载 layer，并在项目中进行引用，官网提供了 demo.html 做为使用案例。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <script src="layer/layer.js"></script>
    </head>
    <body>
        <script>
            layer.open({
                content: 'Hello World！你好，中国！独立使用！'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

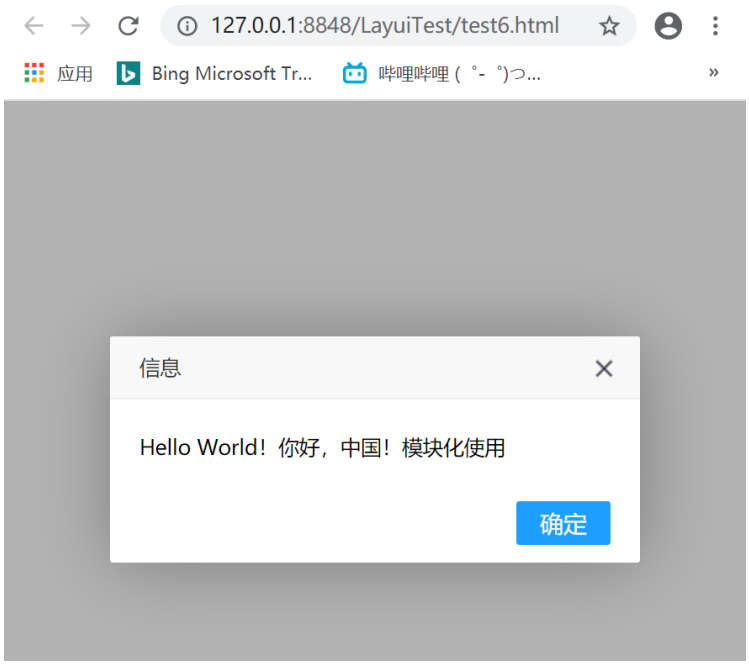


1.18.1.2 模块化使用

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <script>
            var layer = layui.layer;
            layer.open({
                content: 'Hello World！你好，中国！模块化使用'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.18.2 基础参数

基础参数主要指使用弹层时调用方法传入的配置项(方法的参数)，如：

```
layer.open({content: ''});
layer.msg('', {time: 3})
```

其中 content 和 time 即是基础参数，以键值对的形式存在，基础参数可合理应用于任何类型的层中，不需要所有的参数都去配置，大多数参数都是可选的。layer.open()、layer.msg()是内置方法，建议使用 layer.open()方法，使用配置会更加灵活。

1.18.3 type 基本层类型

类型：Number，默认值 0。

layer 提供了 5 种层类型，对 type 参数可传入的值有：

（1）0：信息框，默认值。快速显示文本弹出层的最佳选择。

（2）1：页面层。如果弹出层中显示的内容需要定制，比如 Box 模型这些，建议使用此种方式。

（3）2：iframe 层。在弹出层显示另外一个网页。

（4）3：加载层。比如 ajax 发起请求显示等待状态时可以使用。

（5）4：tips 层。 方便小巧的小提示窗口。

若采用 layer.open({type: x})方式进行调用，则 type 参数为必填项（信息框除外）。

由于参数 type 是需要结合其它属性进行联合使用，所以在后面的章节会结合其它参数来介绍这 5 种层类型的使用。

1.18.4 title 标题

类型：String/Array/Boolean，默认值'信息'。

title 支持三种类型的值：

（1）如果传入的是普通的字符串，如 title:'我是标题'，那么只会改变标题文本。

（2）如果需要自定义标题区域样式，那么可以使用 title:['文本', 'font-size:18px;']，数组第二项可以写任意的 css 样式。

（3）如果不想显示标题栏，可以 title:false。

1.18.4.1 String 类型

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                title: '你好，我是新的标题！',
                content: 'Hello World！你好，中国！<a
href="http://www.baidu.com">百度</a>'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.4.2 Array 类型

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                title: ['你好我是新的标题！ ', 'color:red;font-size:25px'],
                content: 'Hello World！你好，中国！<a
href="http://www.baidu.com">百度</a>'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

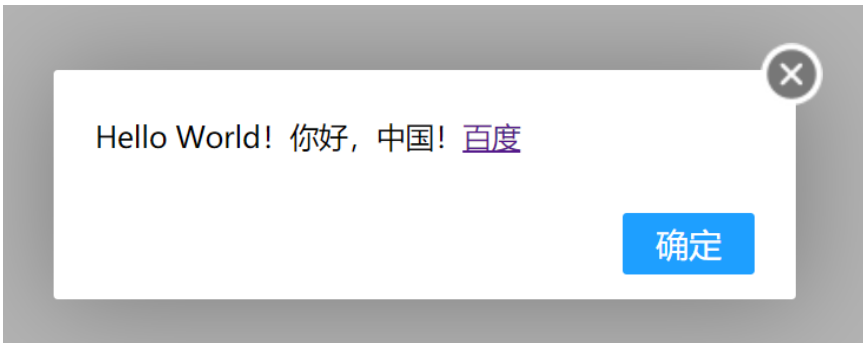
1.18.4.3 Boolean 类型

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
```

```
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                title: ['你好我是新的标题！', 'color:red;font-size:25px'],
                title: false,
                content: 'Hello World！你好，中国！<a
href="http://www.baidu.com">百度</a>'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.18.5 content 内容

类型：String/DOM/Array，默认值''。

content 可传入的值是灵活多变的，不仅可以传入普通的 html 内容，还可以传入 DOM，显示的内容会随着 type 的不同而不同。

注意：type 值使用不当会造成 content 中的内容显示出现错误。

1.18.5.1 String 类型

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
```

```
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                title: '我是标题！',
                content: 'Hello World！你好，中国！<a
href="http://www.baidu.com">百度</a>'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.5.2 DOM 类型

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="div1" style="display: none;">
            <span style="background-color: red;">red</span>
            <span style="background-color: green;">green</span>
            <span style="background-color: blue;">blue</span>
        </div>
        <script>
            var layer = layui.layer;
```

```
            layer.open({
                type: 1,
                title: '我是标题！ ',
                content: $("#div1")
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

通过对 type 传值为 0 和 1 已经看到弹出层在外观上的区别，传值 0 基本属于“拿来主义”，直接显示消息，Box 模型 layer 已经处理完毕，如果传入 1 则弹出层中内容的布局比较原始，Box 模型需要自己来进行定制。

1.18.5.3 Array 类型

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <form class="layui-form" action="">
            <div class="layui-form-item" style="width: 500px;">
                <label class="layui-form-label">输入框</label>
                <div class="layui-input-block">
                    <input type="text" id="username" name="username"
required lay-verify="required" placeholder="请输入标题" autocomplete="off"
                    class="layui-input">
                </div>
            </div>
            <div class="layui-form-item" style="width: 500px;">
                <label class="layui-form-label">密码框</label>
```

```
                <div class="layui-input-block">
                    <input type="password" id="password" name="password"
required lay-verify="required" placeholder="请输入密码"
                     autocomplete="off" class="layui-input">
                </div>
            </div>
        </form>
        <input type="button" id="button1" value="button1"><br />
        <input type="button" id="button2" value="button2"><br />
        <script src="layui/layui.all.js"></script>
        <script>
            $(document).ready(function() {
                $("#button1").click(function() {
                    layer.open({
                        type: 4,
                        content: ['内容不能为空', '#username'] //数组第二项即
吸附元素选择器或者DOM
                    });
                });
                $("#button2").click(function() {
                    layer.open({
                        type: 4,
                        content: ['内容不能为空', '#password'] //数组第二项即
吸附元素选择器或者DOM
                    });
                });
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

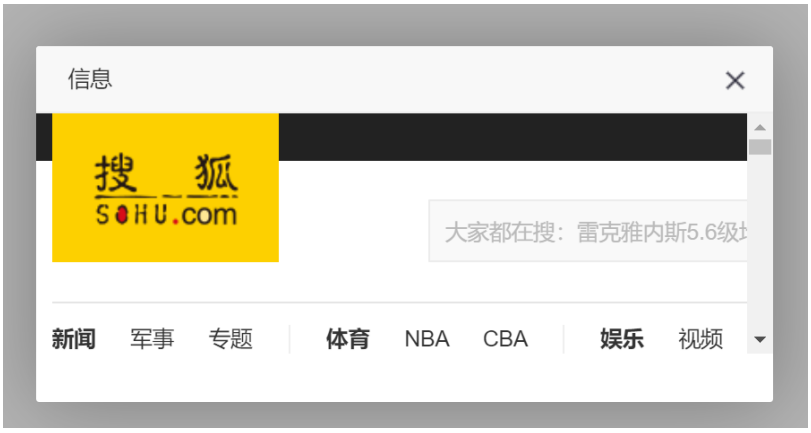
具体 tips 提示层自动关闭或显示位置等功能在后面的章节有介绍。

1.18.5.4 iframe 层的测试

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 2,
                content: 'http://www.sohu.com'
            });
            //这里content是一个URL，如果不想让iframe出现滚动条，
            //还可以content: ['http://www.sohu.com', 'no']
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



本小节只是介绍 iframe 层的基本使用，更多的效果请参看后面的章节。

1.18.6 skin 样式类名

类型：String，默认值''。

skin 不仅允许传入 layer 内置的样式 class 名，还可以传入自定义的 class 名，意味着可

以借助 skin 轻松完成不同的风格定制。目前 layer 内置的 skin 有：

（1）layui-layer-lan

（2）layui-layer-molv

未来还会选择性地内置更多，但更推荐自己来定义。

1.18.6.1 测试默认风格

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                content: '我是中国人'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

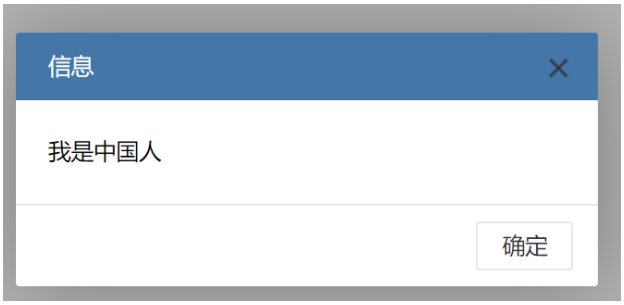
1.18.6.2 测试 layui-layer-lan 风格

测试代码如下：

```
<!DOCTYPE html>
```

```
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                skin: 'layui-layer-lan',
                type: 0,
                content: '我是中国人'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



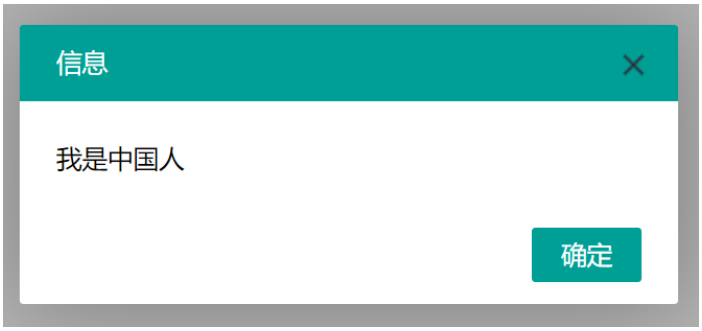
1.18.6.3 测试 layui-layer-molv 风格

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
```

```
            var layer = layui.layer;
            layer.open({
                skin: 'layui-layer-molv',
                type: 0,
                content: '我是中国人'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.18.6.4 测试自定义风格

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
        <style>
            body .demo-class .layui-layer-title {
                background: #c00;
                color: #fff;
                border: none;
            }

            body .demo-class .layui-layer-btn {
                border-top: 1px solid #E9E7E7
            }

            body .demo-class .layui-layer-btn a {
                background: #333;
```

```
            }

            body .demo-class .layui-layer-btn .layui-layer-btn1 {
                background: #999;
            }
        </style>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                skin: 'demo-class',
                type: 0,
                content: '我是中国人'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.7 area 宽高

类型：String/Array，默认值'auto'。

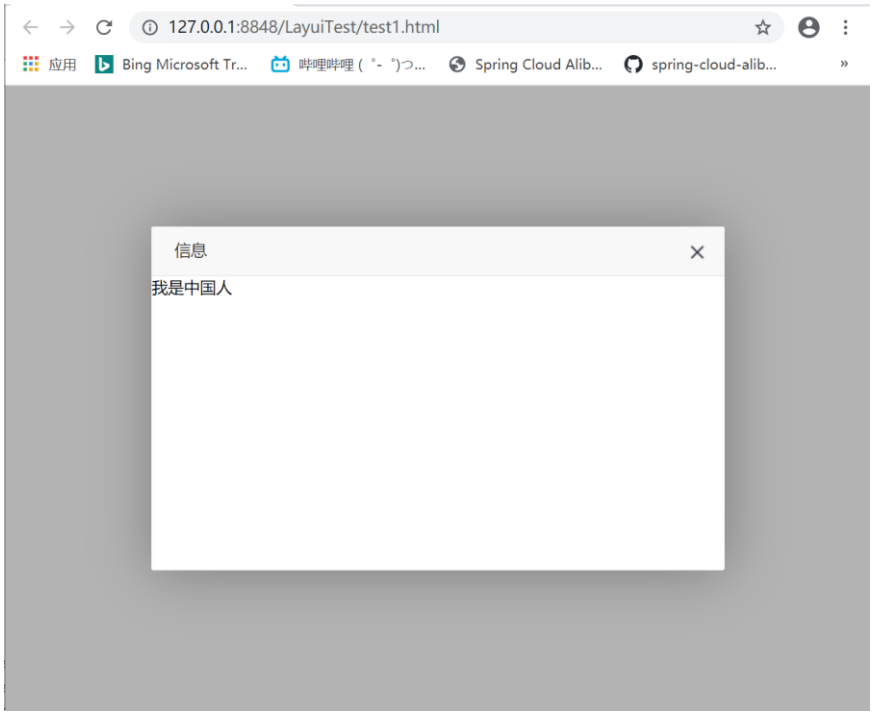
在默认状态下，layer 是宽高都是自适应的，如果想自定义宽度时可以使用 area: '500px'，高度仍然是自适应的，如果宽高都要自定义，可以使用 area: ['500px', '300px']。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
```

```
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                area: ['500px', '300px'],
                content: '我是中国人'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.18.8 offset 坐标

类型：String/Array，默认值“垂直水平居中”。

offset 默认情况下不用设置，但如果不想垂直水平居中，还可以进行如图 1-xx 所示进行配置。

值	备注
offset: 'auto'	默认坐标，即垂直水平居中
offset: '100px'	只定义top坐标，水平保持居中
offset: ['100px', '50px']	同时定义top、left坐标
offset: 't'	快捷设置顶部坐标
offset: 'r'	快捷设置右边缘坐标
offset: 'b'	快捷设置底部坐标
offset: 'l'	快捷设置左边缘坐标
offset: 'lt'	快捷设置左上角
offset: 'lb'	快捷设置左下角
offset: 'rt'	快捷设置右上角
offset: 'rb'	快捷设置右下角

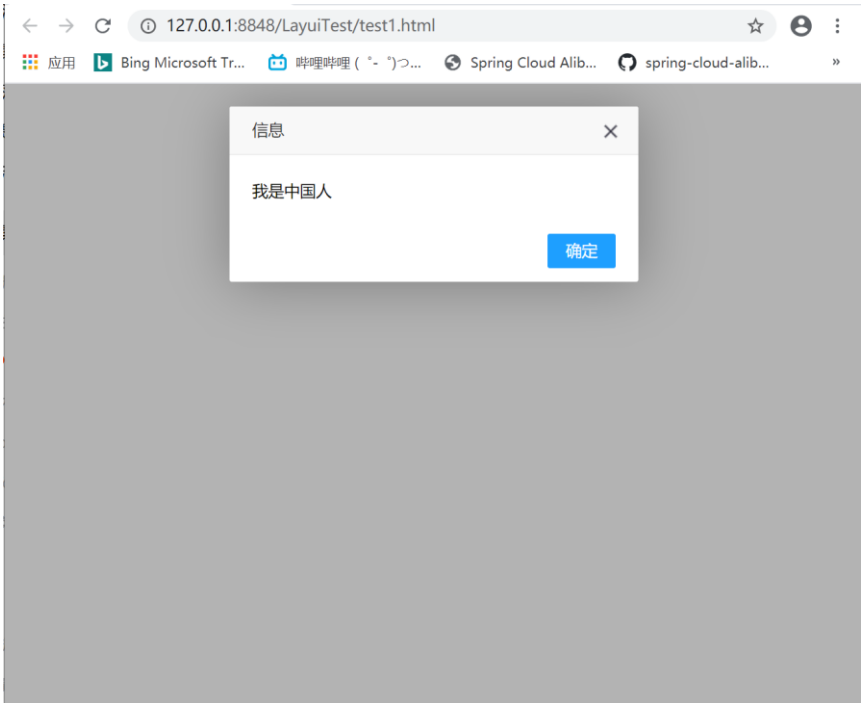
1.18.8.1 测试 offset: '20px'

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                offset: '20px',
                content: '我是中国人'
            });
        </script>
```

```
    </body>
</html>
```

运行效果如图 1-xx 所示。



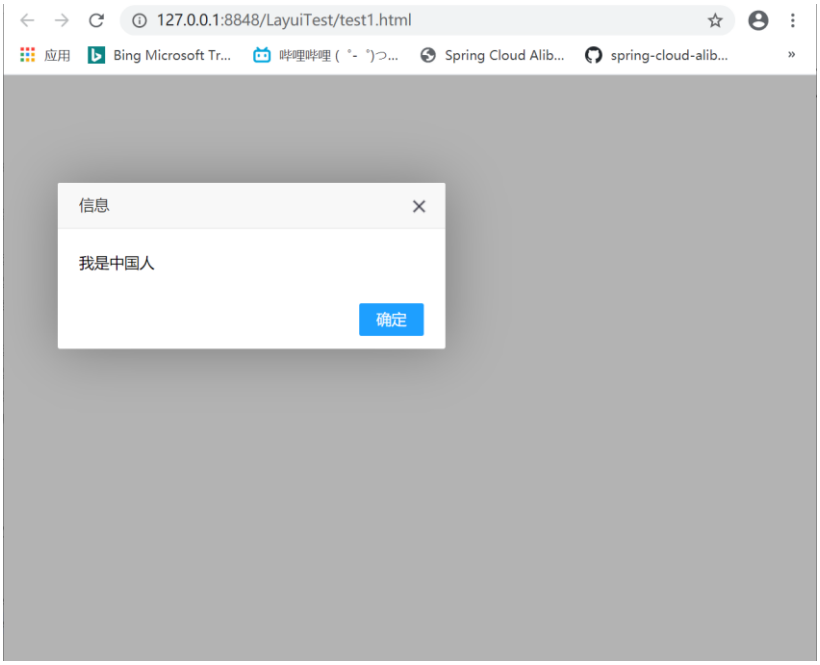
1.18.8.2 测试 offset: ['100px', '50px']

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                offset: ['100px', '50px'],
                content: '我是中国人'
            });
        </script>
```

```
    </body>
</html>
```

运行效果如图 1-xx 所示。



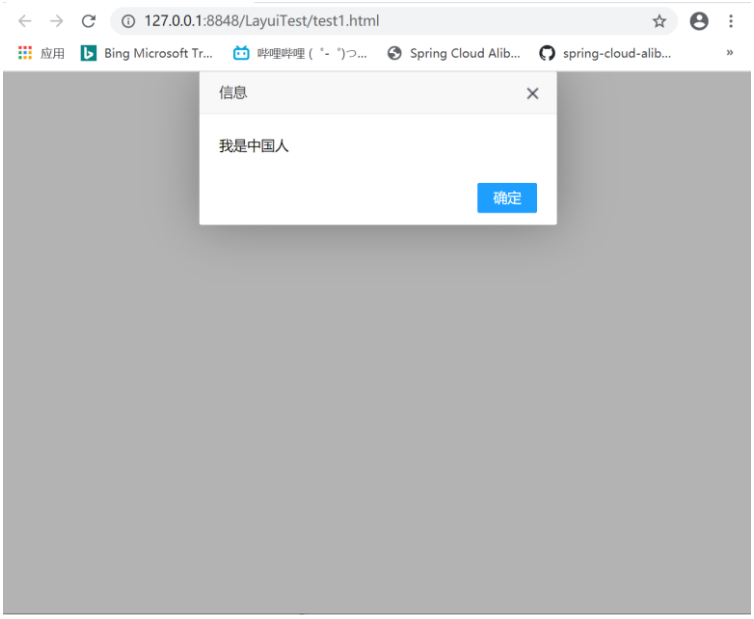
1.18.8.3 测试 offset: 't'

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                offset: 't',
                content: '我是中国人'
            });
        </script>
    </body>
```

```
</html>
```

运行效果如图 1-xx 所示。

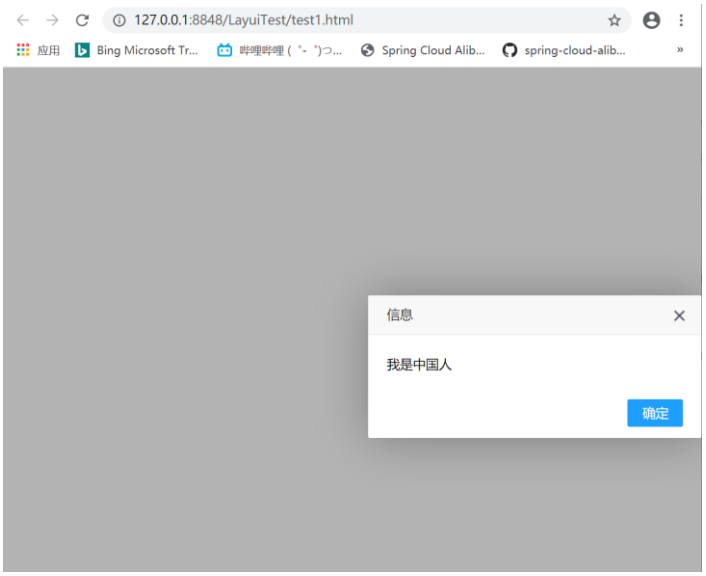


1.18.8.4 测试 offset: 'r'

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                offset: 'r',
                content: '我是中国人'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

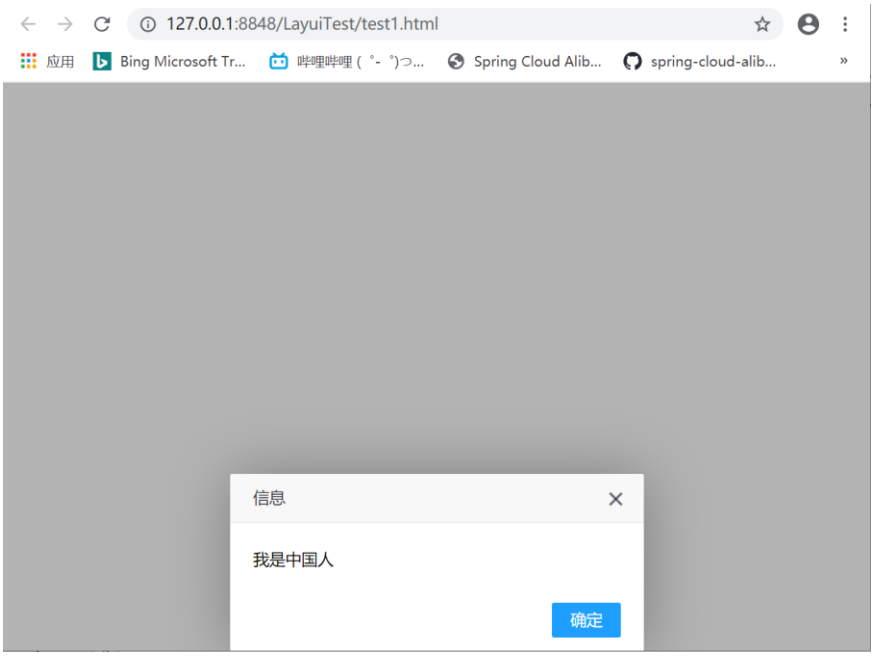


1.18.8.5 测试 offset: 'b'

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                offset: 'b',
                content: '我是中国人'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

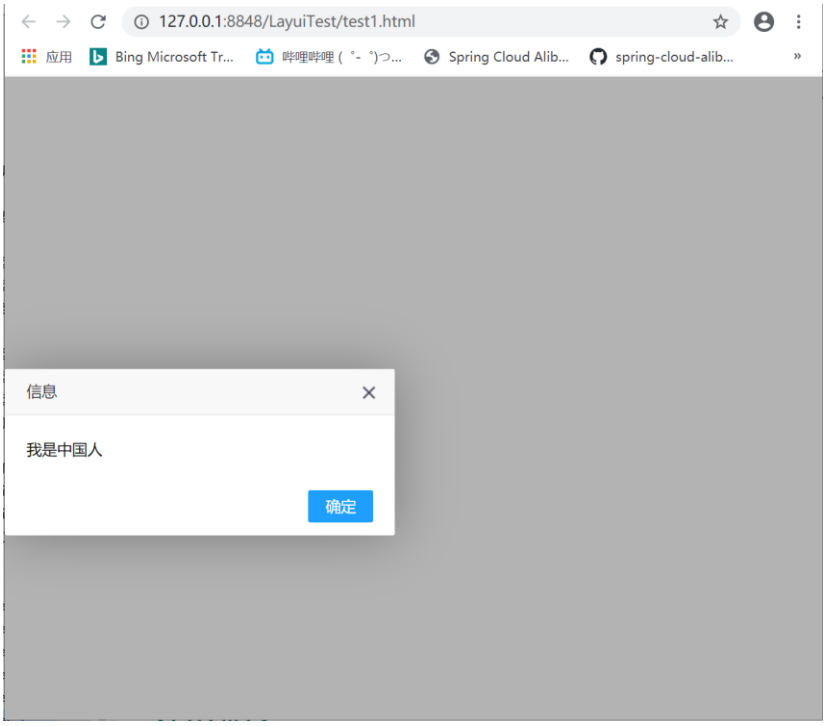


1.18.8.6 测试 offset: 'l'

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                offset: 'l',
                content: '我是中国人'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

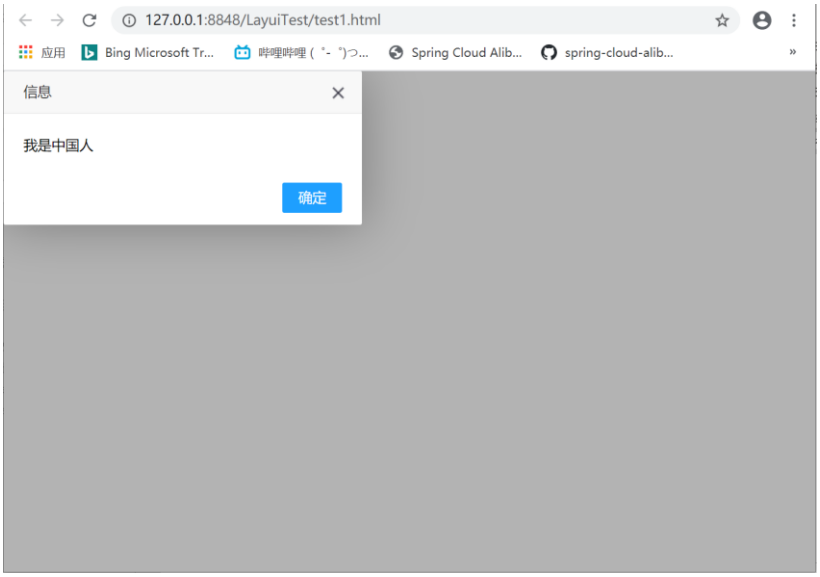


1.18.8.7 测试 offset: 'lt'

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                offset: 'lt',
                content: '我是中国人'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

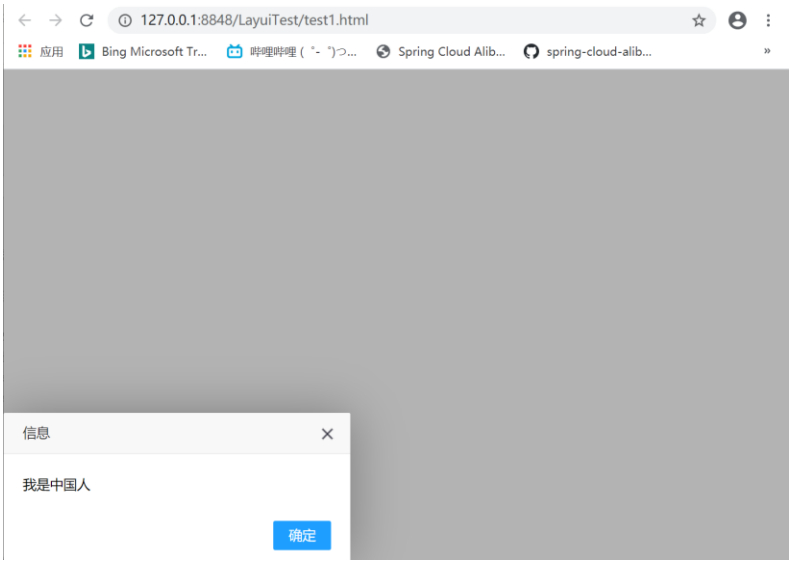


1.18.8.8 测试 offset: 'lb'

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                offset: 'lb',
                content: '我是中国人'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

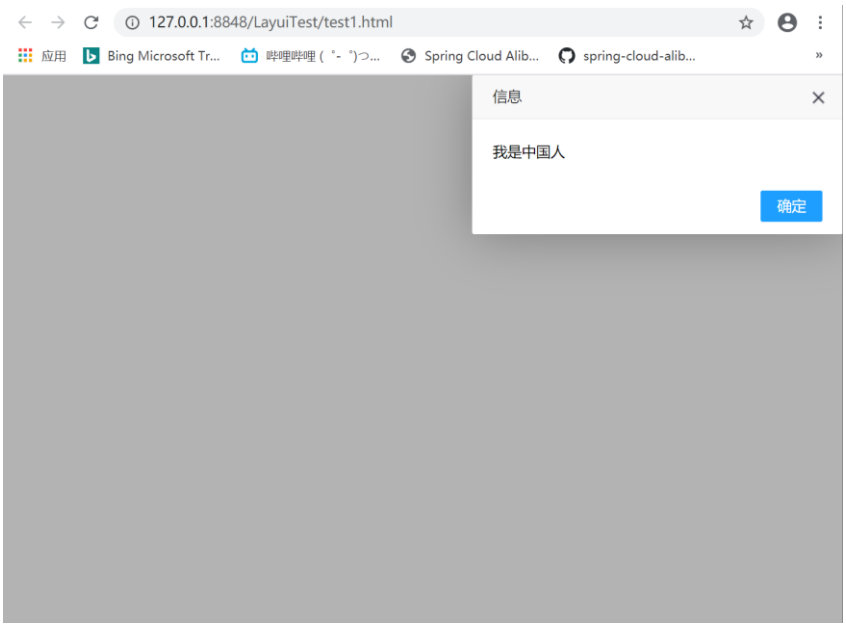


1.18.8.9 测试 offset: 'rt'

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                offset: 'rt',
                content: '我是中国人'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

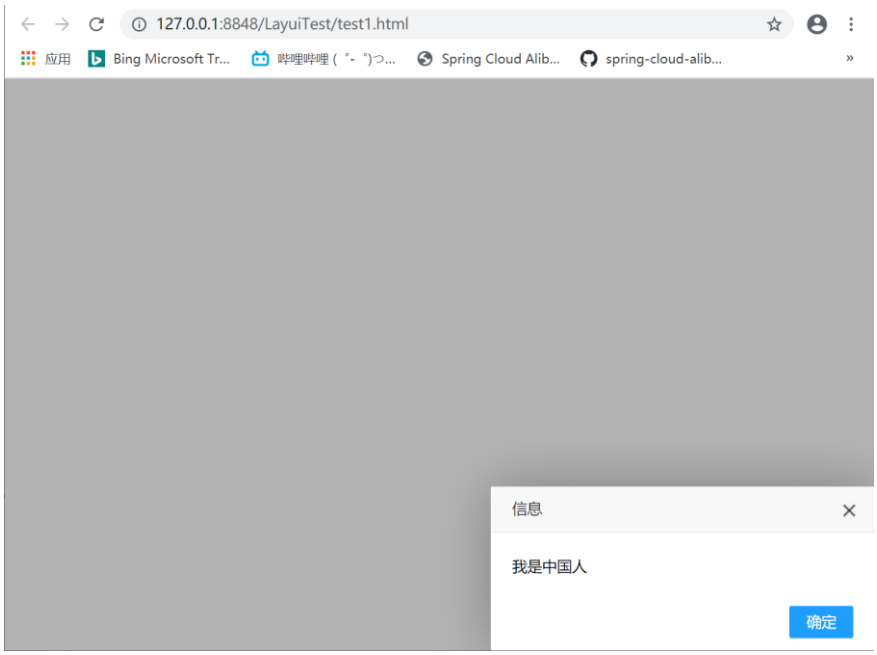


1.18.8.10 测试 offset: 'rb'

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                offset: 'rb',
                content: '我是中国人'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.18.9 icon 图标（信息框和加载层的私有参数）

类型：Number。信息框的默认值是-1，加载层的默认值是 0。
信息框默认不显示图标，当想显示图标时，默认皮肤可以传入 0-6。
如果是加载层，可以传入 0-2。

1.18.9.1 测试信息框 icon: 0

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                icon: 0,
                content: '我是中国人'
            });
```

```
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.9.2 测试信息框 icon: 1

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                icon: 1,
                content: '我是中国人'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.9.3 测试信息框 icon: 2

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                icon: 2,
                content: '我是中国人'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.9.4 测试信息框 icon: 3

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                icon: 3,
                content: '我是中国人'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.9.5 测试信息框 icon: 4

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                icon: 4,
                content: '我是中国人'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.9.6 测试信息框 icon: 5

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                icon: 5,
                content: '我是中国人'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.9.7 测试信息框 icon: 6

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                icon: 6,
                content: '我是中国人'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

图标汇总如图 1-xx 所示。

1.18.9.8 测试加载层 icon: 0

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
```

```
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 3,
                icon: 0
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.9.9 测试加载层 icon: 1

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 3,
                icon: 1
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.9.10 测试加载层 icon: 2

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 3,
                icon: 2
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.10 btn 按钮

类型：String/Array，默认值'确认'。

使用信息框模式时，btn 默认是一个“确认”按钮，其它层类型则默认不显示，加载层和 tips 层则无效。当想自定义一个按钮时，可以使用 btn: '我知道了'，当要定义两个按钮时可以使用 btn: ['yes', 'no']。当然，也可以定义更多按钮，比如：btn: ['按钮 1', '按钮 2', '按钮 3', …]，按钮 1 的回调是 yes，而从按钮 2 开始，则回调为 btn2: function(){}，以此类推。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                content: "test text",
                btn: ['按钮一', '按钮二', '按钮三'],
                yes: function(index, layero) {
                    //按钮【按钮一】的回调
                    layer.close(index);
                },
                btn2: function(index, layero) {
                    //按钮【按钮二】的回调
                    //return false 开启该代码可禁止点击该按钮关闭
                },
                btn3: function(index, layero) {
                    //按钮【按钮三】的回调
                    return false; //开启该代码可禁止点击该按钮关闭
                },
                cancel: function() {
                    //右上角关闭回调
                    return false; //开启该代码可禁止点击该按钮关闭
                }
            });
        </script>
    </body>
```

```
</html>
```

运行效果如图 1-xx 所示。

1.18.11 btnAlign 按钮排列

类型：String，默认值为 r。

可以快捷定义按钮的排列位置，btnAlign 的默认值为 r，即右对齐。该参数可支持的赋值如图 1-xx 所示。

值	备注
btnAlign: 'l'	按钮左对齐
btnAlign: 'c'	按钮居中对齐
btnAlign: 'r'	按钮右对齐。默认值，不用设置

1.18.11.1 测试 btnAlign: 'l'

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
```

```
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                content: "test text",
                btn: ['按钮一', '按钮二', '按钮三'],
                yes: function(index, layero) {
                    //按钮【按钮一】的回调
                    layer.close(index);
                },
                btn2: function(index, layero) {
                    //按钮【按钮二】的回调
                    //return false 开启该代码可禁止点击该按钮关闭
                },
                btn3: function(index, layero) {
                    //按钮【按钮三】的回调
                    return false; //开启该代码可禁止点击该按钮关闭
                },
                cancel: function() {
                    //右上角关闭回调
                    return false; //开启该代码可禁止点击该按钮关闭
                },
                btnAlign: 'l'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.11.2 测试 btnAlign: 'c'

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                content: "test text",
                btn: ['按钮一', '按钮二', '按钮三'],
                yes: function(index, layero) {
                    //按钮【按钮一】的回调
                    layer.close(index);
                },
                btn2: function(index, layero) {
                    //按钮【按钮二】的回调
                    //return false 开启该代码可禁止点击该按钮关闭
                },
                btn3: function(index, layero) {
                    //按钮【按钮三】的回调
                    return false; //开启该代码可禁止点击该按钮关闭
                },
                cancel: function() {
                    //右上角关闭回调
                    return false; //开启该代码可禁止点击该按钮关闭
                },
                btnAlign: 'c'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.11.3 测试 btnAlign: 'r'

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                content: "test text",
                btn: ['按钮一', '按钮二', '按钮三'],
                yes: function(index, layero) {
                    //按钮【按钮一】的回调
                    layer.close(index);
                },
                btn2: function(index, layero) {
                    //按钮【按钮二】的回调
                    //return false 开启该代码可禁止点击该按钮关闭
                },
                btn3: function(index, layero) {
                    //按钮【按钮三】的回调
                    return false; //开启该代码可禁止点击该按钮关闭
                },
                cancel: function() {
                    //右上角关闭回调
```

```
                return false; //开启该代码可禁止点击该按钮关闭
            },
            btnAlign: 'r'
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。

1.18.12 closeBtn 关闭按钮

类型：String/Boolean，默认值为 1。

layer 提供了两种风格的关闭按钮，可通过配置 1 和 2 来展示。如果不显示，则 closeBtn: 0 或 closeBtn: false。

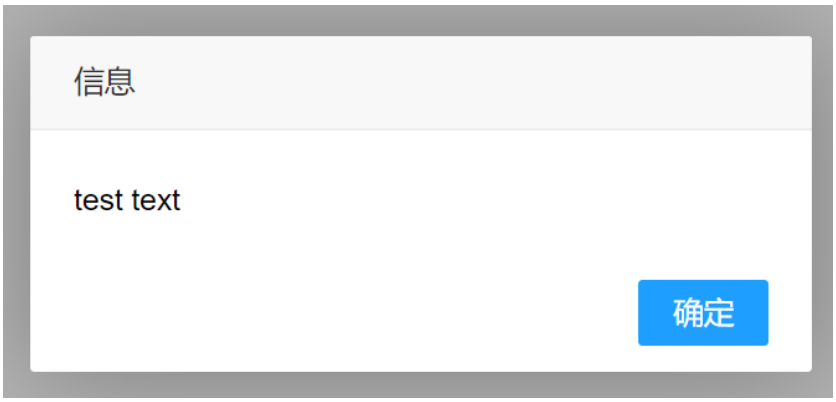
1.18.12.1 测试 closeBtn: 0

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
```

```
            content: "test text",
            closeBtn: 0
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。

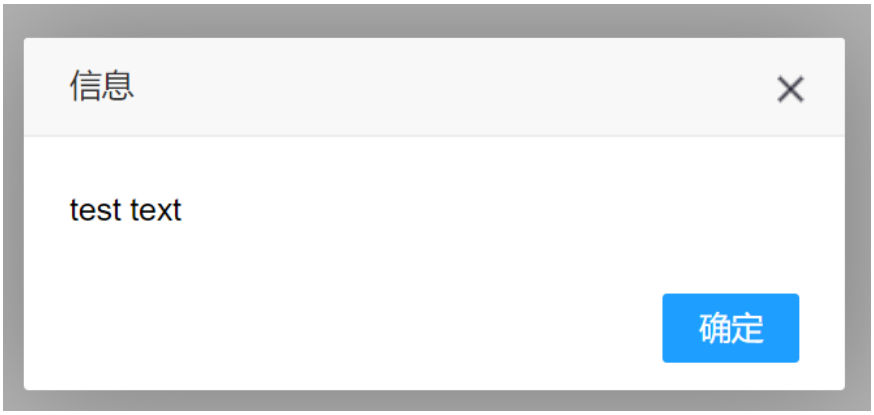


1.18.12.2 测试 closeBtn: 1

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                content: "test text",
                closeBtn: 1
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

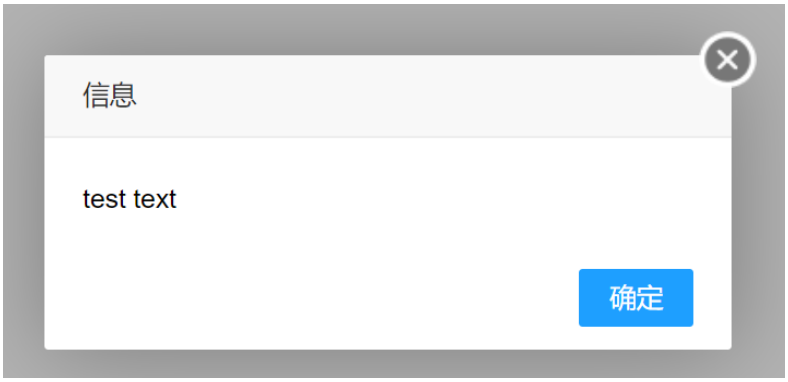


1.18.12.3 测试 closeBtn: 2

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                content: "test text",
                closeBtn: 2
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.18.13 shade 遮罩

类型：Number/Array，默认值 0.3。

即弹层外区域。默认是 0.3 透明度的黑色背景（'#000'）。如果想定义别的颜色，可以 shade: [0.8, '#393D49']，如果不想显示遮罩，可以 shade: 0。

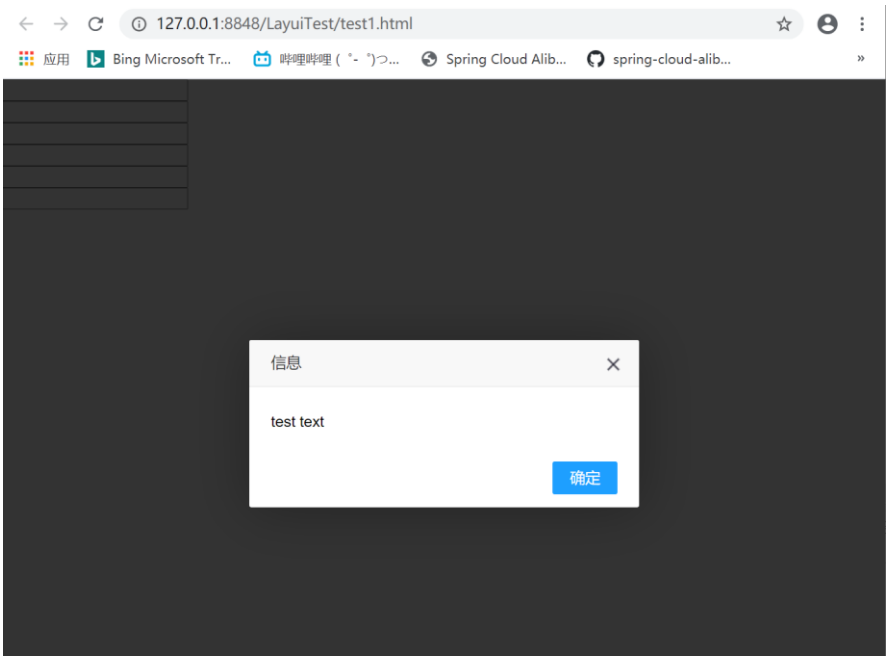
1.18.13.1 测试 Number

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                content: "test text",
                shade: 0.8
            });
        </script>
```

```
    </body>
</html>
```

运行效果如图 1-xx 所示。



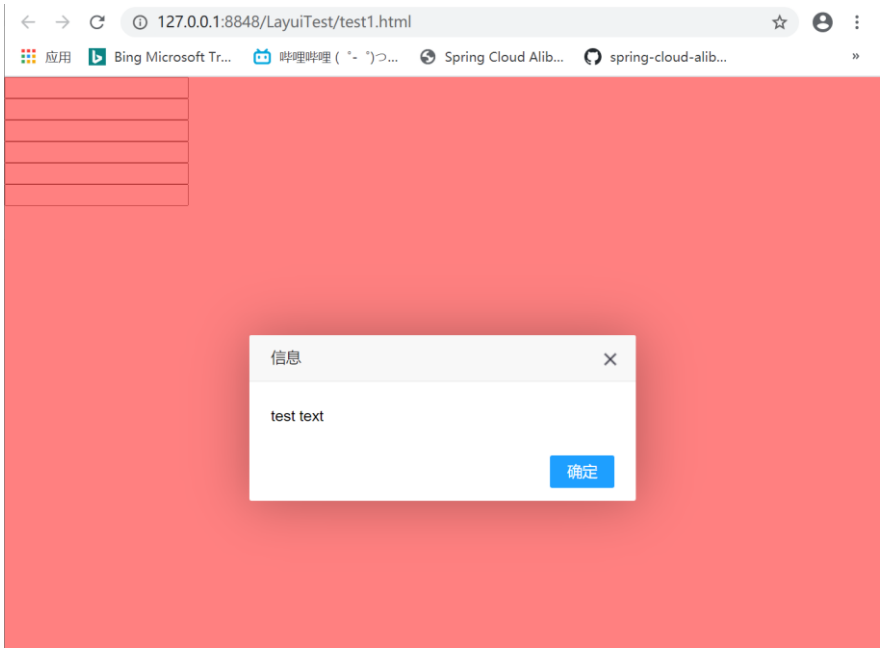
1.18.13.2 测试 Array

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
```

```
                content: "test text",
                shade: [0.5, 'red']
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



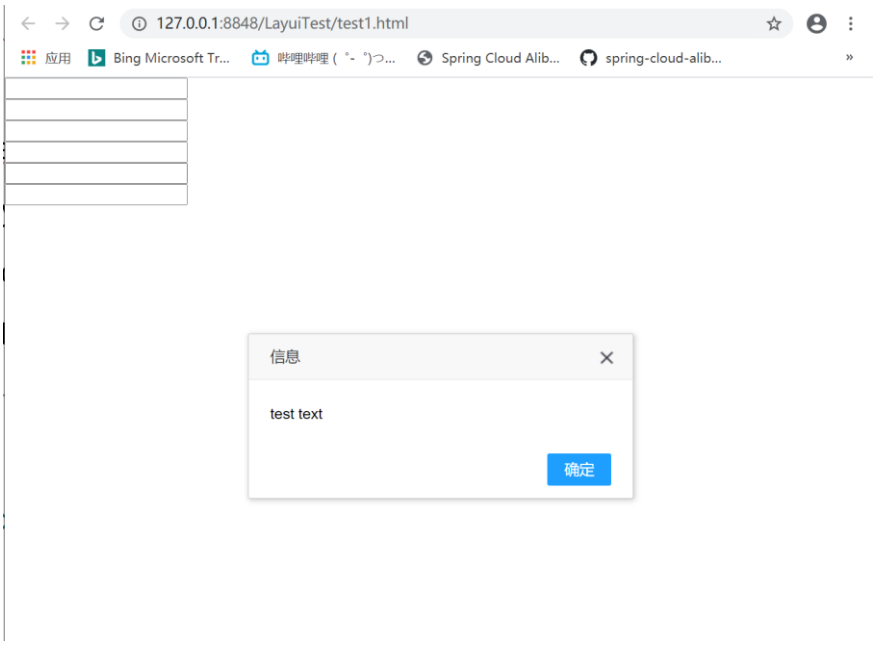
1.18.13.3 不显示遮罩

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
```

```
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                content: "test text",
                shade: 0
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.18.14 shadeClose 是否点击遮罩关闭

类型：Boolean，默认值 false。
如果 shade 是存在的，那么可以设置 shadeClose 来控制点击弹层外区域关闭。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
```

```
    </head>
    <body>
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                content: "test text",
                shadeClose: true
            });
        </script>
    </body>
</html>
```

点击遮罩关闭弹层。

1.18.15 time 自动关闭所需毫秒

类型：Number，默认值 0。

默认不会自动关闭。当想自动关闭时，可以设置 time: 5000，代表 5 秒后自动关闭，注意单位是毫秒（1 秒=1000 毫秒）

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
        <input type="text" /><br />
```

```
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                content: "test text",
                time: 3000
            });
        </script>
    </body>
</html>
```

3 秒后弹层自动关闭。

1.18.16 id 用于控制弹层唯一标识

类型：String，默认值为空字符。

设置该值后，不管是什么类型的层，都只允许同时弹出一个。一般用于页面层和 iframe 层模式。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <input type="button" id="button1" value="button1" /><br />
        <input type="button" id="button2" value="button2" /><br />
        <script src="layui/layui.all.js"></script>
        <script>
            layui.use('layer', function() {
                var layer = layui.layer;

            });
        </script>
        <script>
            $(document).ready(function() {
                $("#button1").click(function() {
                    layer.open({
                        type: 1,
```

```
                content: "test text 1",
                offset: 'l'
            });
            layer.open({
                type: 1,
                content: "test text 2"
            });
        });

        $("#button2").click(function() {
            layer.open({
                type: 1,
                content: "test text 3",
                id: '101',
                offset: 'l'
            });
            layer.open({
                type: 1,
                content: "test text 4",
                id: '101'
            });
        });
    });
</script>
</body>
</html>
```

点击 Button1 同时弹出 2 个层，点击 Button2 同时只能弹出 1 个层，因为 id 值一样。

1.18.17 anim 弹出动画

类型：Number，默认值 0。

出场动画全部采用 CSS3。这意味着除了 ie6-9，其它所有浏览器都是支持的。目前 anim 可支持的动画类型有 0-6 ，如果不想显示动画，设置 anim: -1 即可。

动画效果如图 1-xx 所示。

值	备注
anim: 0	平滑放大。默认
anim: 1	从上掉落
anim: 2	从最底部往上滑入
anim: 3	从左滑入
anim: 4	从左翻滚
anim: 5	渐显
anim: 6	抖动

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                content: "test text 1",
                anim: 6 //属性值可以写0-6
            });
        </script>
    </body>
</html>
```

请自行更改动画种类值为 0-6，一共 7 种动画。

1.18.18 isOutAnim 关闭退场动画

类型：Boolean，默认值 true。

默认情况下，关闭层时会有一个退场过度动画。如果不想开启，设置 isOutAnim: false 即可。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                content: "test text 1",
                anim: 6,
                isOutAnim: false
            });
        </script>
    </body>
</html>
```

1.18.19 maxmin 最大化最小化

类型：Boolean，默认值 false。

该参数值对 type:1 和 type:2 有效。默认不显示最大化最小化按钮。需要显示配置 maxmin: true 即可。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
```

```
                type: 1,
                content: "test text 1",
                maxmin: true
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.20 fixed 固定

类型：Boolean，默认值 true。
即鼠标滚动时，层是否固定在可视区域。如果不想，设置 fixed: false 即可。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br />
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                content: "test text 1",
```

```
                fixed: false
            });
        </script>
    </body>
</html>
```

鼠标滚轮滚动时，弹出层的位置也随之改变。

1.18.21 resize 是否允许拉伸

类型：Boolean，默认值 true。

默认情况下，可以在弹层右下角拖动来拉伸尺寸。如果对指定的弹层屏蔽该功能，设置 false 即可。该参数对 loading、tips 层无效。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                content: "test text 1",
                resize: false
            });
        </script>
    </body>
</html>
```

弹出层的大小不可以改变。

1.18.22 resizing 监听窗口拉伸动作

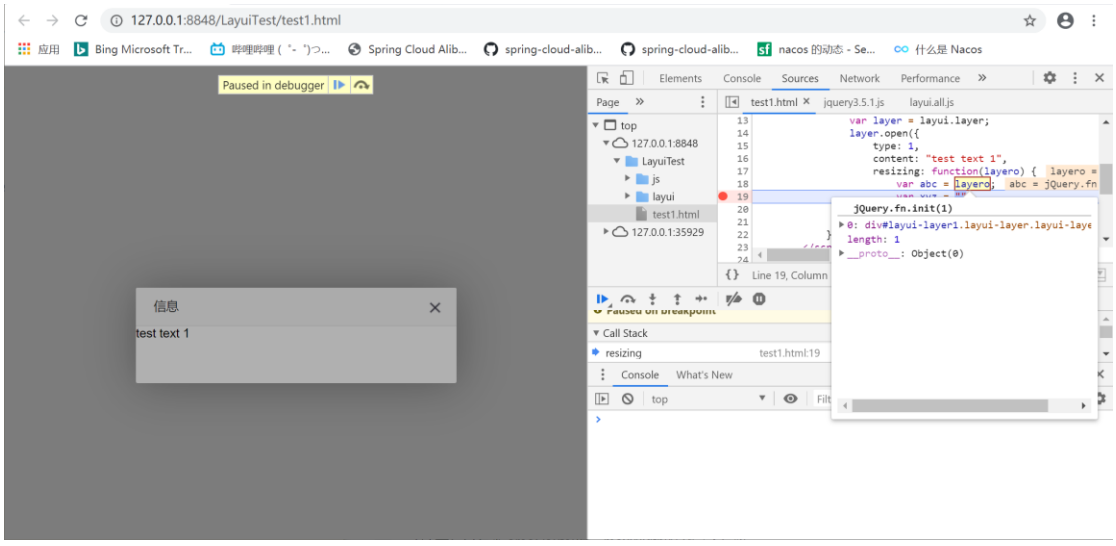
类型：Function，默认值 null。

当拖拽弹层右下角进行尺寸调整时，如果设置了该回调，则会执行。回调返回一个参数：当前层的 DOM 对象。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                content: "test text 1",
                resizing: function(layero) {
                    var abc = layero;
                    var xyz = "";
                }
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.18.23 scrollbar 是否允许浏览器出现滚动条

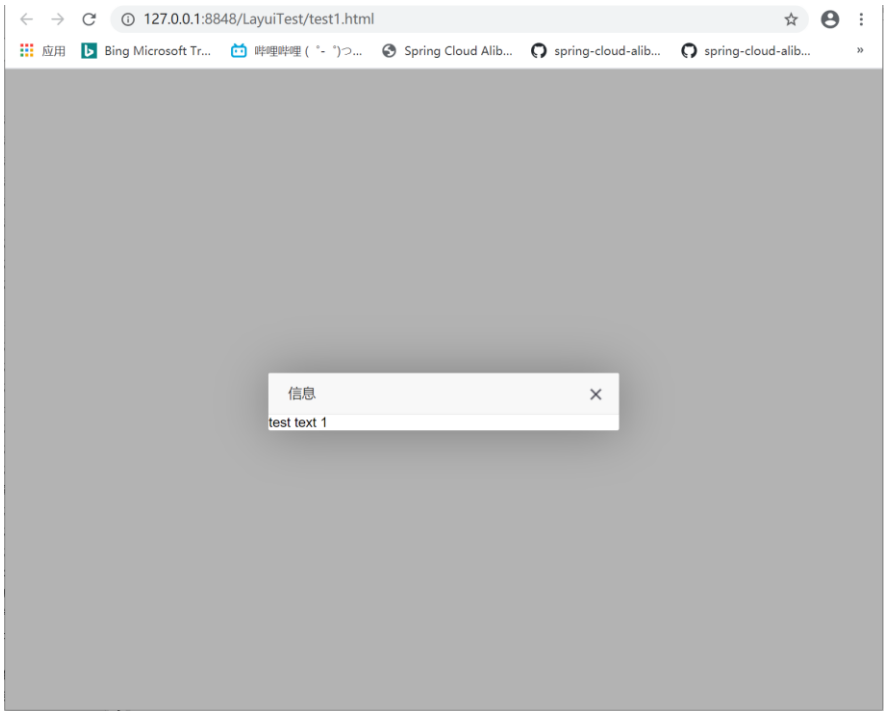
类型：Boolean，默认值 true。

默认允许浏览器滚动，如果设置 scrollbar: false，则屏蔽。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br />
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                content: "test text 1",
                scrollbar: false
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.18.24 maxWidth 最大宽度

类型：Number，默认值 360px。

请注意：只有当 area: 'auto'时，maxWidth 的设置才有效。

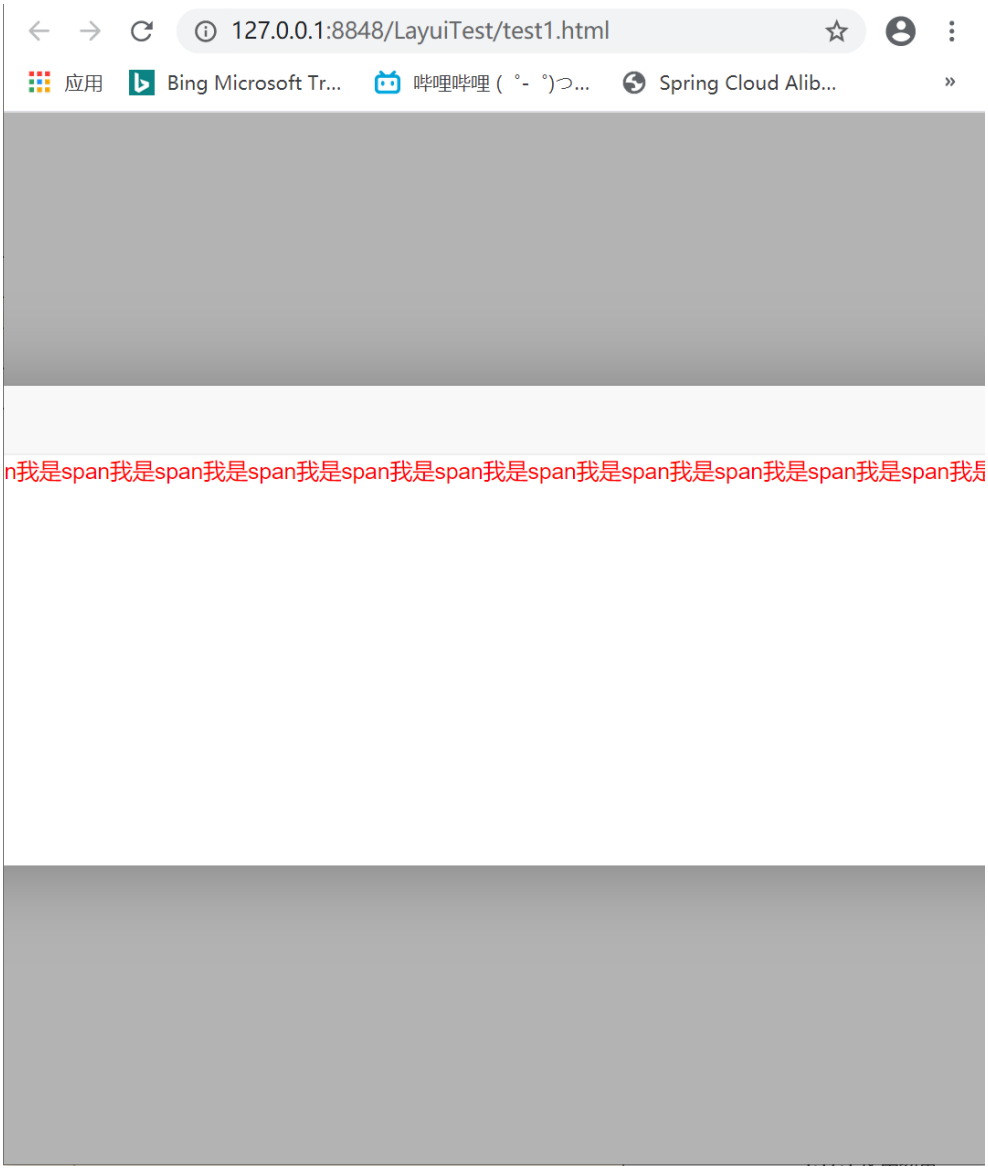
一般在布局时，不想要元素的宽度限定死，并且想要它的实际宽度随其本身内容自适应，但又不想宽度过大破坏整体布局，这时候就会应用 max-width 限制元素的最大宽度，元素实际宽度在 0~max-width 之间。具有自动换行的效果。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                content: "begin<span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
```

```
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span>end",
                area: ['2000px', '300px'],
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



宽高都固定了，并且看不到 begin 和 end 字符串信息，过多的信息全被隐藏了，这就是固定宽高的缺点。

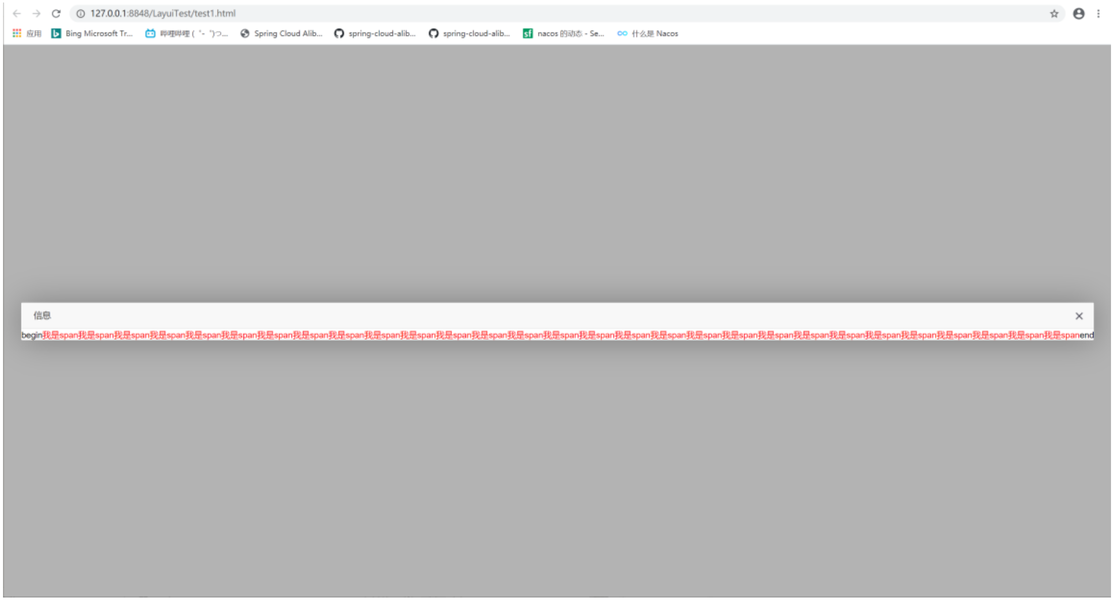
使用属性 maxWidth 除了支持最大宽度外还可以实现宽度自适应。

测试代码如下：

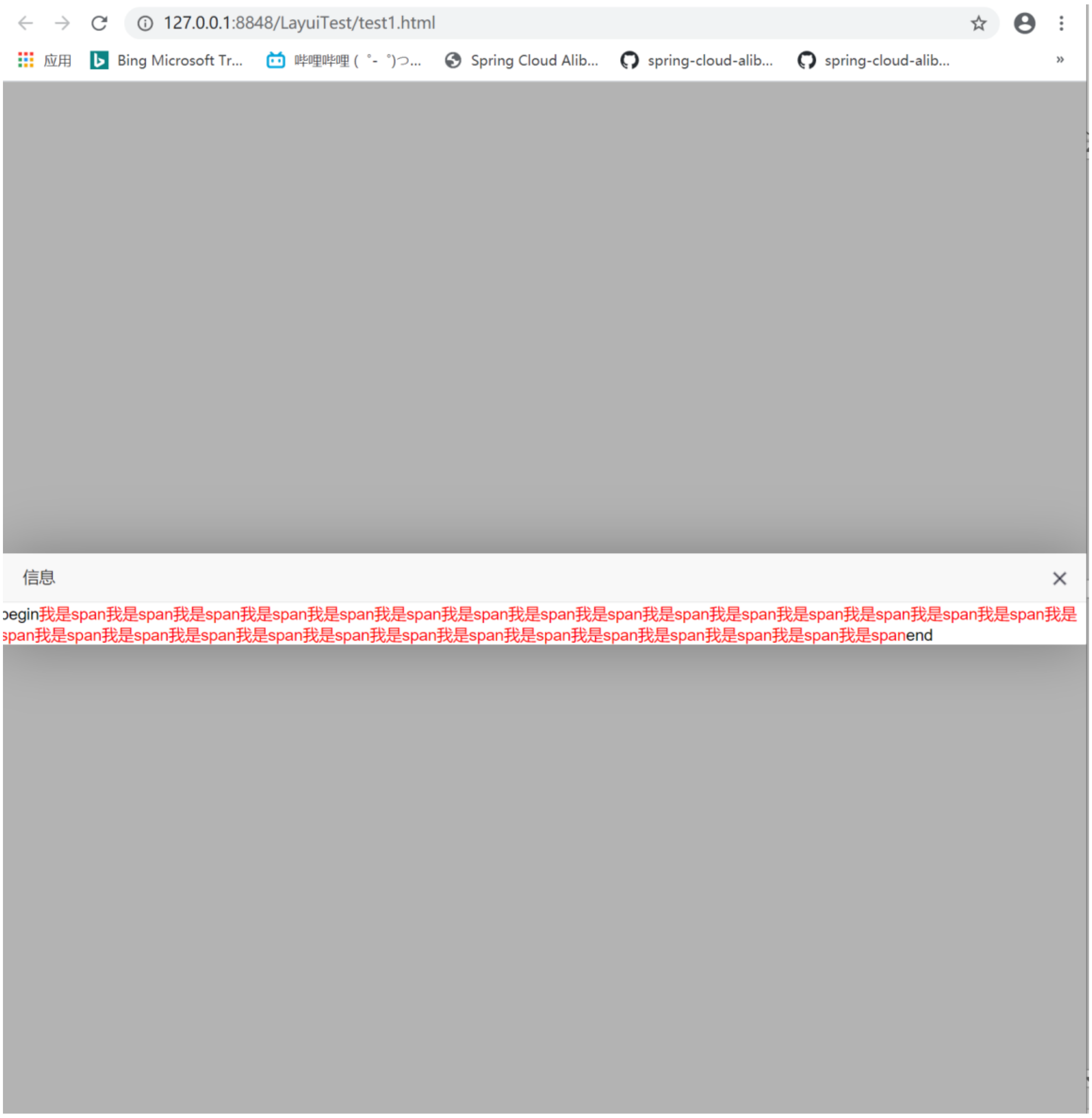
```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
```

```
            layer.open({
                type: 1,
                content: "begin<span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span>end",
                area: 'auto',
                maxWidth: 2000
            });
        </script>
    </body>
</html>
```

最大宽度显示效果如图 1-xx 所示。



并且还具有宽度自适应的效果，如图 1-xx 所示。



浏览器在任何宽度下都可以把信息显示完整。

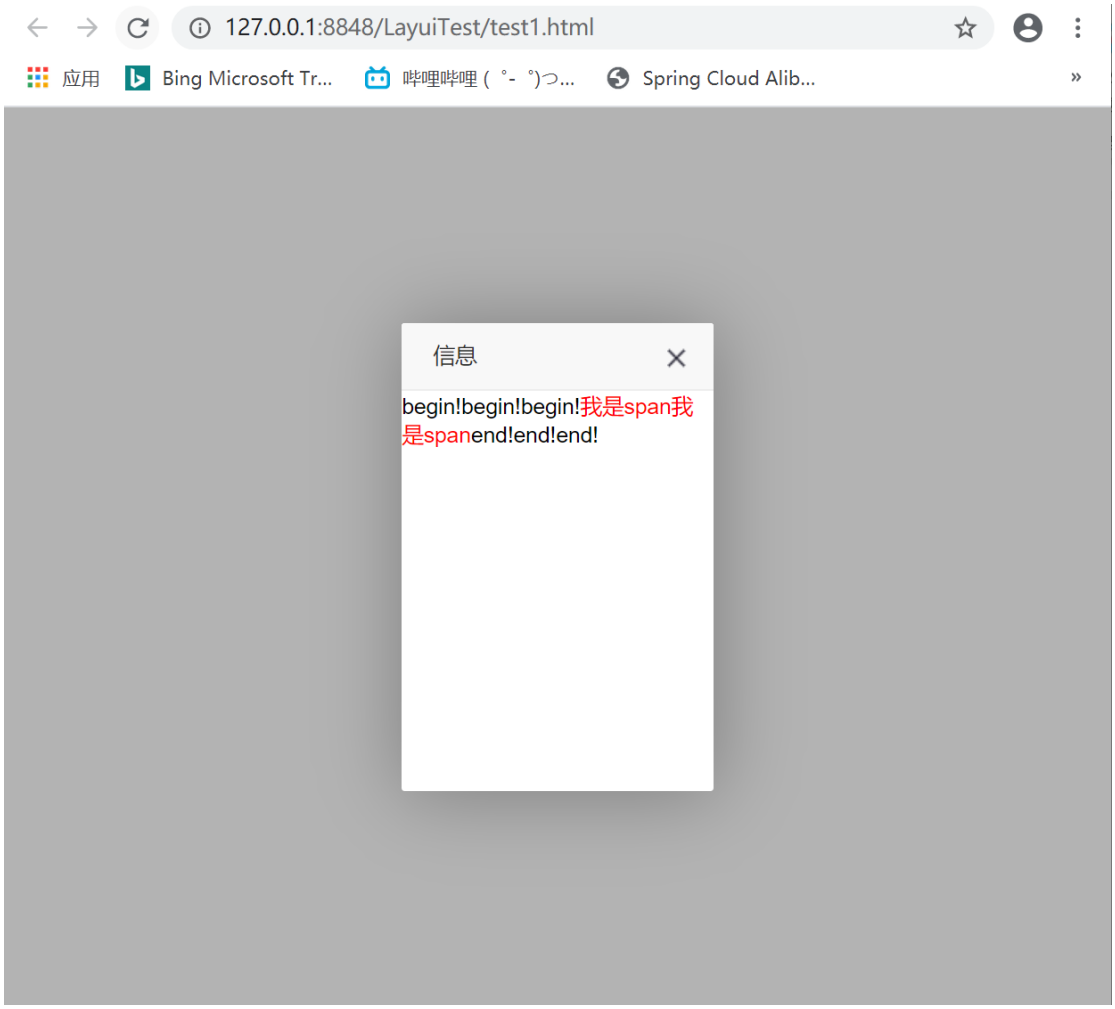
1.18.25 maxHeight 最大高度

类型：Number，默认值无。
请注意：只有当高度自适应时，maxHeight 的设置才有效。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                content: "begin!begin!begin!<span style='color:red'>我是
span</span><span style='color:red'>我是span</span>end!end!end!",
                area: ['200px', '300px']
            });
        </script>
    </body>
</html>
```

固定宽高的运行效果如图 1-xx 所示。



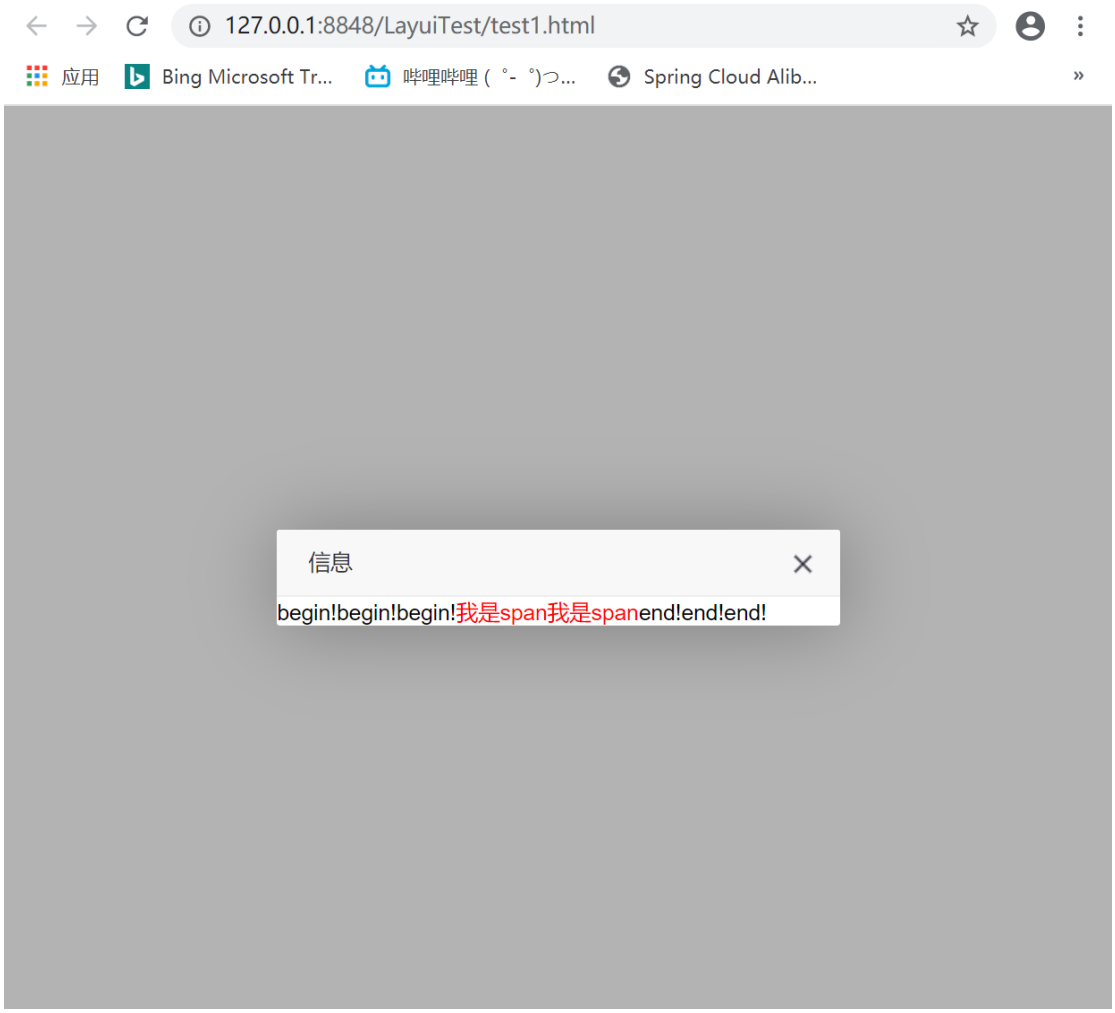
在弹出层的下面出现了大面积空白，影响界面美观性，属性 maxHeight 支持高度自适应。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                content: "begin!begin!begin!<span style='color:red'>我是
```

```
span</span><span style='color:red'>我是span</span>end!end!end!",
                area: 'auto',
                maxHeight: 400
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



通过使用属性 maxHeight 后，弹出层下面的大面积空白消失了，说明属性 maxHeight 支持高度自适应。

如果内容过多，最大高度 maxHeight 还有效吗？

测试代码如下：

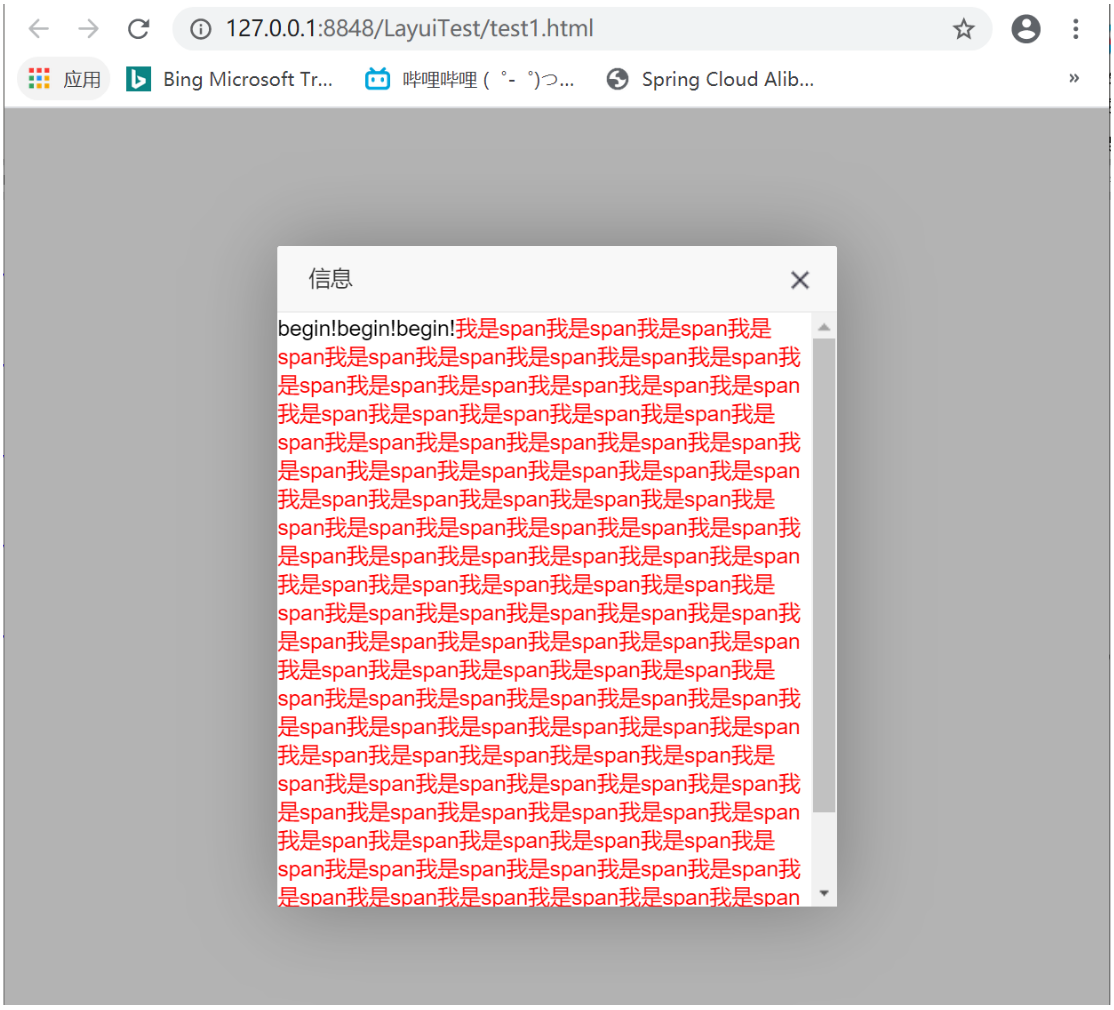
```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
```

```
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                content: "begin!begin!begin!<span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
```

```
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
```

```
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span><span
style='color:red'>我是span</span><span style='color:red'>我是
span</span><span style='color:red'>我是span</span>end!end!end!",
                area: 'auto',
                maxHeight: 400
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



内容超过 400px 后出现垂直滚动条，最大高度是 400px。

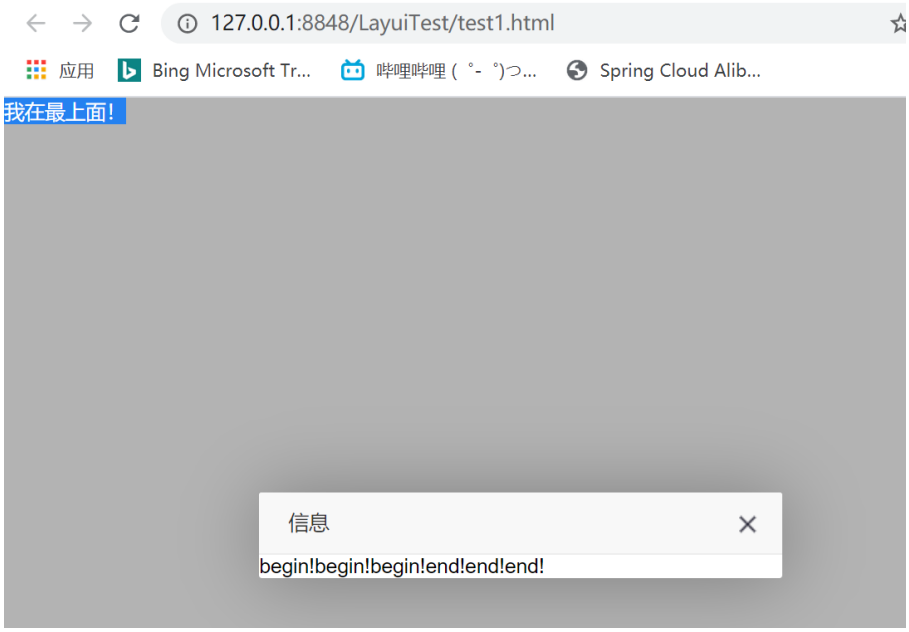
1.18.26 zIndex 层叠顺序

默认值 19891014（作者贤心生日）。
一般用于解决和其它组件的层叠冲突。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div style="position: absolute; z-index: 19891015;">我在最上面!
</div>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                content: "begin!begin!begin!end!end!end!",
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

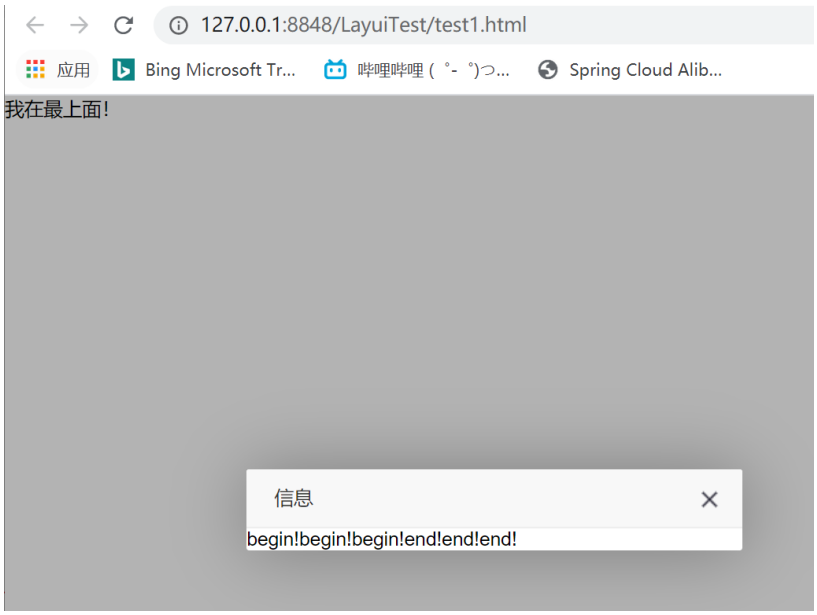


自定义 div 的 z-index: 19891015 值大于 19891014，所以文字可以选中，说明自定义 div 在遮罩层之上。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div style="position: absolute; z-index: 100;">我在最上面！ </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                content: "begin!begin!begin!end!end!end!",
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



文字“我在最上面！”选中不了了，因为在遮罩层的下面，被遮罩层覆盖了。

1.18.27 move 触发拖动的元素

类型：String/DOM/Boolean，默认值'.layui-layer-title'。

默认是触发标题区域实现拖拽。如果想单独定义，指向元素的选择器或者 DOM 即可。如 move: '.mine-move'。还配置设置 move: false 来禁止拖拽。

1.18.27.1 测试 String

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                content: "<span class='myspanclass'>我的span</span>",
                move: '.myspanclass'
            });
        </script>
    </body>
</html>
```

1.18.27.2 测试 DOM

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div id="mydiv" style="display: none;">
```

```
            <div>我是div</div>
        </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                content: $("#mydiv"),
                move: $("#mydiv div")
            });
        </script>
    </body>
</html>
```

1.18.27.3 禁用 move

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                content: "<span style='color:red'>我是span</span>",
                move: false
            });
        </script>
    </body>
</html>
```

1.18.28 moveOut 是否允许拖拽到窗口外

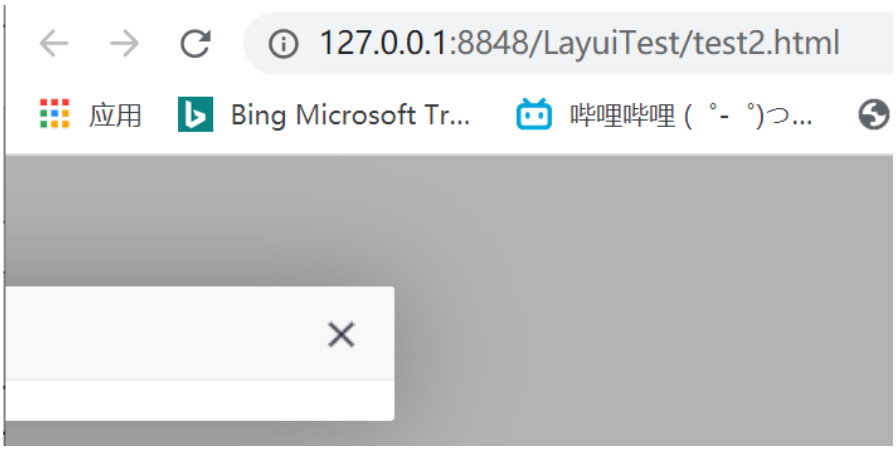
类型：Boolean，默认值 false。

默认只能在窗口内拖拽，如果想拖到窗外，那么设置 moveOut: true 即可

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                content: "<span style='color:red'>我是span</span>",
                moveOut: true
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.18.29 moveEnd 拖动完毕后的回调方法

类型：Function，默认值 null。

默认不会触发 moveEnd，如果需要可以设置 moveEnd: function(layero){}即可。其中 layero 为当前层的 DOM 对象

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                content: "<span style='color:red'>我是span</span>",
                moveEnd: function(layero) {
                    alert('拖动结束！ ');
                }
            });
        </script>
    </body>
</html>
```

1.18.30 tips 方向和颜色

类型：Number/Array，默认值 2。

tips 层的私有参数。支持上右下左四个方向，通过 1-4 进行方向设置。如 tips: 3 则表示在元素的下面出现。有时还可能会定义一些颜色，可以设置 tips: [1, '#c00']。

1.18.30.1 测试 1

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <form class="layui-form" action="">
            <div class="layui-form-item" style="width: 500px;margin:
```

```
400px;"">
                <label class=" layui-form-label">输入框</label>
                <div class="layui-input-block">
                    <input type="text" id="username" name="username"
required lay-verify="required" placeholder="请输入标题" autocomplete="off"
                           class="layui-input">
                </div>
            </div>
        </form>
        <script src="layui/layui.all.js"></script>
        <script>
            var form = layui.form;
            var layer = layui.layer;
            layer.open({
                type: 4,
                tips: [1, 'red'],
                content: ['内容不能为空', '#username']
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.30.2 测试 2

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
```

```
    <body>
        <form class="layui-form" action="">
            <div class="layui-form-item" style="width: 500px;margin:
400px;"">
                <label class=" layui-form-label">输入框</label>
                <div class="layui-input-block">
                    <input type="text" id="username" name="username"
required lay-verify="required" placeholder="请输入标题" autocomplete="off"
                     class="layui-input">
                </div>
            </div>
        </form>
        <script src="layui/layui.all.js"></script>
        <script>
            var form = layui.form;
            var layer = layui.layer;
            layer.open({
                type: 4,
                tips: [2, 'red'],
                content: ['内容不能为空', '#username']
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.30.3 测试 3

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
```

```
        <form class="layui-form" action="">
            <div class="layui-form-item" style="width: 500px;margin:
400px;"">
                <label class=" layui-form-label">输入框</label>
                <div class="layui-input-block">
                    <input type="text" id="username" name="username"
required lay-verify="required" placeholder="请输入标题" autocomplete="off"
                     class="layui-input">
                </div>
            </div>
        </form>
        <script src="layui/layui.all.js"></script>
        <script>
            var form = layui.form;
            var layer = layui.layer;
            layer.open({
                type: 4,
                tips: [3, 'red'],
                content: ['内容不能为空', '#username']
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.30.4 测试 4

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
```

```
    </head>
    <body>
        <form class="layui-form" action="">
            <div class="layui-form-item" style="width: 500px;margin:
400px;"">
                <label class=" layui-form-label">输入框</label>
                <div class="layui-input-block">
                    <input type="text" id="username" name="username"
required lay-verify="required" placeholder="请输入标题" autocomplete="off"
                     class="layui-input">
                </div>
            </div>
        </form>
        <script src="layui/layui.all.js"></script>
        <script>
            var form = layui.form;
            var layer = layui.layer;
            layer.open({
                type: 4,
                tips: [4, 'red'],
                content: ['内容不能为空', '#username']
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.31 tipsMore 是否允许同时显示多个 tips

类型：Boolean，默认值 false。
允许多个意味着不会销毁之前的 tips 层。通过 tipsMore: true 开启

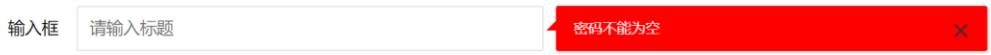
测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
```

```
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <form class="layui-form" action="">
            <div class="layui-form-item" style="width: 500px;margin:
400px;"">
                <label class=" layui-form-label">输入框</label>
                <div class="layui-input-block">
                    <input type="text" id="username" name="username"
required lay-verify="required" placeholder="请输入标题" autocomplete="off"
                     class="layui-input">
                </div>
            </div>
            <div class="layui-form-item" style="width: 500px;margin:
400px;"">
                <label class=" layui-form-label">输入框</label>
                <div class="layui-input-block">
                    <input type="text" id="password" name="password"
required lay-verify="required" placeholder="请输入标题" autocomplete="off"
                     class="layui-input">
                </div>
            </div>
        </form>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            var form = layui.form;
            layer.open({
                type: 4,
                tips: [2, 'red'],
                content: ['账号不能为空', '#username'],
                tipsMore: true,
                shade: false,
                fixed: false
            });
            layer.open({
                type: 4,
                tips: [2, 'red'],
                content: ['密码不能为空', '#password'],
                tipsMore: true,
                shade: false,
                fixed: false
            });
```

```
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.18.32 success 层弹出后的成功回调方法

类型：Function，默认值 null。

当需要在层创建完毕时执行一些语句，可以通过该回调。success 会携带两个参数，分别是当前层 DOM 和当前层索引。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
```

```
                type: 1,
                content: "<span style='color:red'>我是span</span>",
                success: function(layero, index) {
                    alert(layero + "  " + index);
                }
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.18.33 yes 确定按钮回调方法

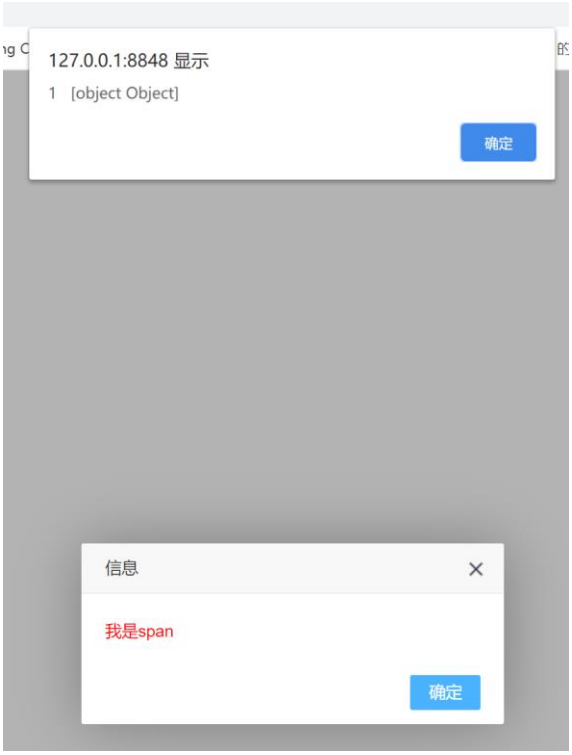
类型：Function，默认值 null。
该回调携带两个参数，分别为当前层索引、当前层 DOM 对象。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                content: "<span style='color:red'>我是span</span>",
                yes: function(index, layero) {
                    alert(index + "   " + layero);
```

```
                layer.close(index); //如果设置了yes回调，需进行手工关闭
            }
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.18.34 cancel 右上角关闭按钮触发的回调

类型：Function，默认值 null。

该回调携带两个参数，分别为：当前层索引参数（index）、当前层的 DOM 对象（layero），默认会自动触发关闭。如果不想关闭，return false 即可。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
```

```
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                content: "<span style='color:red'>我是span</span>",
                cancel: function(index, layero) {
                    if (confirm('确定要关闭么')) { //只有当点击confirm框的确
定时，该层才会关闭
                        layer.close(index)
                    }
                    return false;
                }
            });
        </script>
    </body>
</html>
```

1.18.35 end 层销毁后触发的回调

类型：Function，默认值 null。
无论是确认还是取消，只要层被销毁了，end 回调都会执行，不携带任何参数。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 0,
                content: "<span style='color:red'>我是span</span>",
                end: function() {
                    alert('销毁了！ ');
                }
            });
```

```
        </script>
    </body>
</html>
```

1.18.36 full/min/restore 分别代表最大化、最小化、还原后触发的回调

类型：Function，默认值 null。
携带一个参数，即当前层 DOM。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.open({
                type: 1,
                content: "<span style='color:red'>我是span</span>",
                maxmin: true,
                full: function(layero) {
                    alert('full! ');
                },
                min: function(layero) {
                    alert('min! ');
                },
                restore: function(layero) {
                    alert('restore! ');
                }
            });
        </script>
    </body>
</html>
```

1.18.37 layer.ready(callback)初始化就绪

由于 layer 内置了轻量级加载器，所以根本不需要单独引入 css 等文件，但是加载总是需要过程和时间的，当在页面一打开就要执行弹层时，最好是将弹层放入 ready 方法中。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            layer.ready(function() {
                layer.msg('很高兴一开场就见到你');
            });
        </script>
    </body>
</html>
```

1.18.38 layer.alert(content, options, yes)普通信息框

用于对用户造成比较强烈的关注。options 是可选参数，yes 是回调方法。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <input type="button" id="button1" value="button1"><br />
        <input type="button" id="button2" value="button2"><br />
        <input type="button" id="button3" value="button3"><br />
        <script src="layui/layui.all.js"></script>
```

```
        <script>
            var layer = layui.layer;

            $("#button1").click(function() {
                layer.alert('只想简单的提示');
            });

            $("#button2").click(function() {
                layer.alert('加了个图标', {
                    icon: 1
                });
            });

            $("#button3").click(function() {
                //这时如果你也还想执行yes回调，可以放在第三个参数中。
                layer.alert('有了回调', function(index) {
                    //do something
                    alert("点击了确定！");
                    layer.close(index);
                });
            });
        </script>
    </body>
</html>
```

1.18.39 layer.confirm(content, options, yes, cancel)询问框

系统的 confirm 具有阻塞特性，而 layer.confirm(content, options, yes, cancel)需要把交互的语句放在回调体中。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <input type="button" id="button1" value="button1"><br />
        <input type="button" id="button2" value="button2"><br />
        <script src="layui/layui.all.js"></script>
```

```
        <script>
            var layer = layui.layer;

            $("#button1").click(function() {
                layer.confirm('is not?', {
                    icon: 3,
                    title: '提示'
                }, function(index) {
                    //do something
                    alert(index);
                    layer.close(index);
                });
            });

            $("#button2").click(function() {
                layer.confirm('is not?', function(index) {
                    //do something
                    alert(index);
                    layer.close(index);
                });
            });
        </script>
    </body>
</html>
```

1.18.40 layer.msg(content, options, end)提示框

开发 msg 方法的目的是想将其打造成露脸率最高的提示框，而事实上的确也在大量地使用它，因为它简单，而且占据很少的面积，默认还会 3 秒后自动消失。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <input type="button" id="button1" value="button1"><br />
        <input type="button" id="button2" value="button2"><br />
        <input type="button" id="button3" value="button3"><br />
```

```
        <input type="button" id="button4" value="button4"><br />
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;

            $("#button1").click(function() {
                layer.msg('只想弱弱提示');
            });

            $("#button2").click(function() {
                layer.msg('有表情地提示', {
                    icon: 6
                });
            });

            $("#button3").click(function() {
                layer.msg('关闭后想做些什么', function() {
                    //do something
                });
            });

            $("#button4").click(function() {
                layer.msg('同上', {
                    icon: 1,
                    time: 2000 //2秒关闭（如果不配置，默认是3秒）
                }, function() {
                    //do something
                });
            });
        </script>
    </body>
</html>
```

1.18.41 layer.load(icon, options)加载层

type:3 的深度定制。load 并不需要传太多的参数，如果不喜欢默认的加载风格，icon 支持传入 0-2，0 是默认值。另外特别注意一点：load 默认是不会自动关闭的，因为一般会在 ajax 回调体中关闭它。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
```

```
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <input type="button" id="button1" value="button1"><br />
        <input type="button" id="button2" value="button2"><br />
        <input type="button" id="button3" value="button3"><br />
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;

            $("#button1").click(function() {
                var index = layer.load();
                setTimeout(function() {
                    layer.close(index);
                }, 3000)
            });

            $("#button2").click(function() {
                var index = layer.load(1); //换了种风格
                setTimeout(function() {
                    layer.close(index);
                }, 3000)
            });

            $("#button3").click(function() {
                var index = layer.load(2, {
                    time: 10 * 1000
                }); //又换了种风格，并且设置最长等待10秒
            });
        </script>
    </body>
</html>
```

1.18.42 layer.tips(content, follow, options)层

type:4 的深度定制。默认是在元素右边弹出

测试代码如下：

```
<!DOCTYPE html>
<html>
```

```
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <form class="layui-form" action="">
            <div class="layui-form-item" style="width: 500px;margin:
400px;"">
                        <label class=" layui-form-label">输入框</label>
                <div class="layui-input-block">
                    <input type="text" id="username" name="username"
required lay-verify="required" placeholder="请输入标题" autocomplete="off"
                     class="layui-input">
                </div>
            </div>
        </form>
        <script src="layui/layui.all.js"></script>
        <input type="button" id="button1" value="button1"><br />
        <input type="button" id="button2" value="button2"><br />
        <script src="layui/layui.all.js"></script>
        <script>
            var form = layui.form;
            var layer = layui.layer;

            $(document).ready(function() {
                $("#button1").click(function() {
                    layer.tips('只想提示地精准些', '#username');
                });

                $('#username').on('click', function() {
                    var that = this;
                    layer.tips('只想提示地精准些', that); //在元素的事件回调
体中，follow直接赋予this即可
                });

                $("#button2").click(function() {
                    layer.tips('在右面', '#username', {
                        tips: 2
                    });
                });
            });
        </script>
```

```
    </body>
</html>
```

1.18.43 layer.close(index)关闭特定层

想关闭当前页的某个层时，需要调用 close(index)方法，而获得 index 代码如下：
var index = layer.open();
var index = layer.alert();
var index = layer.load();
var index = layer.tips();
每一种弹层调用方式都会返回一个 index 值，代表弹层的唯一标识，再结合 close()方法：
layer.close(index);
实现弹层的关闭。

如果想关闭最新弹出的层，直接获取 layer.index 即可，代码如下：
layer.close(layer.index);
它获取的 index 值始终是最新弹出的某个层的 index 值，值是由 layer 内部动态递增计算的。

当在 iframe 页面关闭自身时使用如下代码：
var index = parent.layer.getFrameIndex(window.name);
先得到当前 iframe 层的索引，然后再执行关闭方法：
parent.layer.close(index);

1.18.43.1 layer.close(index)关闭特定层

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            var index = layer.open({
                title: false,
                type: 1,
                closeBtn: 0,
```

```
                content: "<div id='mydiv'
style='background-color:blue;color:white'>倒计时：10秒！</div>",
                success: function(layero, index) {
                    var runTime = 0;
                    var intervalHandler = setInterval(function() {
                        if (runTime == 10) {
                            clearInterval(intervalHandler);
                            layer.close(index);
                        } else {
                            $(layero).find("#mydiv").text("倒计时：" + (10 -
runTime) + "秒！");
                        }
                        runTime++;
                    }, 1000);
                }
            });
        </script>
    </body>
</html>
```

1.18.43.2 关闭 iframe 弹出层

文件 iframe.html 代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <input type="button" value="关闭iframe弹出层" id="closeSelf">
        <script src="layui/layui.all.js"></script>
        <script>
            $(document).ready(function() {
                $("#closeSelf").click(function() {
                    var index = parent.layer.getFrameIndex(window.name);
                    parent.layer.close(index);
                });
            });
        </script>
    </body>
</html>
```

运行 html 代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            var index = layer.open({
                type: 2,
                content: "iframe.html"
            });
        </script>
    </body>
</html>
```

1.18.44 layer.closeAll(type)关闭所有层

如果不想获取 index 而实现关闭弹层时，可以使用 closeAll 方法。如果不指定层类型的话，它会销毁当前页面中所有的 layer 层。当然，如果只想关闭某个类型的层，那么可以使用如下代码：

```
layer.closeAll(); //疯狂模式，关闭所有层
layer.closeAll('dialog'); //关闭信息框
layer.closeAll('page'); //关闭所有页面层
layer.closeAll('iframe'); //关闭所有的 iframe 层
layer.closeAll('loading'); //关闭加载层
layer.closeAll('tips'); //关闭所有的 tips 层
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
```

```
    <body>
        <input type="button" value="closeAll" id="closeAll">
        <form class="layui-form" action="">
            <div class="layui-form-item" style="width: 500px;margin:
400px;"">
                <label class=" layui-form-label">输入框</label>
                <div class="layui-input-block">
                    <input type="text" id="username" name="username"
required lay-verify="required" placeholder="请输入标题" autocomplete="off"
                     class="layui-input">
                </div>
            </div>
            <div class="layui-form-item" style="width: 500px;margin:
400px;"">
                <label class=" layui-form-label">输入框</label>
                <div class="layui-input-block">
                    <input type="text" id="password" name="password"
required lay-verify="required" placeholder="请输入标题" autocomplete="off"
                     class="layui-input">
                </div>
            </div>
        </form>
        <script src="layui/layui.all.js"></script>
        <script>
            var form = layui.form;
            var layer = layui.layer;
            layer.open({
                type: 4,
                tips: [2, 'red'],
                content: ['账号不能为空', '#username'],
                tipsMore: true,
                shade: false,
                fixed: false
            });
            layer.open({
                type: 4,
                tips: [2, 'red'],
                content: ['密码不能为空', '#password'],
                tipsMore: true,
                shade: false,
                fixed: false
            });

            $("#closeAll").click(function() {
```

```
                layer.closeAll();
            });
        </script>
    </body>
</html>
```

1.18.45 layer.style(index, cssStyle)重新定义层 content 中的样式

该方法对 loading 层和 tips 层无效。参数 index 为层的索引，cssStyle 允许传入任意的 css 属性：

```
layer.style(index, {
  width: '1000px',
  top: '10px'
});
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            var index = layer.open({
                type: 1,
                content: "我是弹出层1",
                shade: false
            });
            setTimeout(function() {
                layer.style(index, {
                    width: '800px',
                    backgroundColor: 'green',
                });
            }, 3000);
        </script>
    </body>
</html>
```

1.18.46 layer.title(title, index)改变层的标题

使用方式：layer.title('标题变了', index)

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;
            var index = layer.open({
                type: 1,
                content: "我是弹出层1",
                shade: false
            });
            setTimeout(function() {
                layer.title("zzzzzzzzzzzzzzzzzzzzzzzzzzz", index)
            }, 3000);
        </script>
    </body>
</html>
```

1.18.47 layer.getChildFrame(selector, index)获取 iframe 页的 DOM

当试图在当前页获取 iframe 页的 DOM 元素时，可以用此方法。selector 即 iframe 页的选择器

文件 iframe.html 代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script>
            function testMethod() {
                document.getElementById("divX").innerText =
```

```
document.getElementById("divX").innerText + "__testMethod方法添加的文本__";
            }
        </script>
    </head>
    <body>
        <div id="divX" style="background-color: red;color:white">我是Div，我在另外一个页面中！</div>
    </body>
</html>
```

运行文件代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            layui.use('layer', function() {
                var layer = layui.layer;

                var index = layer.open({
                    type: 2,
                    content: 'iframe.html',
                    success: function(layero, index) {
                        var body = layer.getChildFrame('body', index);
                        $(body).find("div").append("__这条信息来自于父页面！");
                        var iframeWin = window[layero.find('iframe')[0]['name']]; //得到iframe页的窗口对象，执行iframe页的方法：iframeWin.method();
                        iframeWin.testMethod();
                    }
                });
            });
        </script>
    </body>
</html>
```

1.18.48 layer.getFrameIndex(windowName)获取特定 iframe 层的索引

一般用于在 iframe 页关闭自身时用到。
此方法在前面章节已经介绍完毕。

1.18.49 layer.iframeAuto(index)指定 iframe 层自适应

调用该方法时，iframe 层的高度会重新进行适应

文件 iframe.html 测试代码如下：

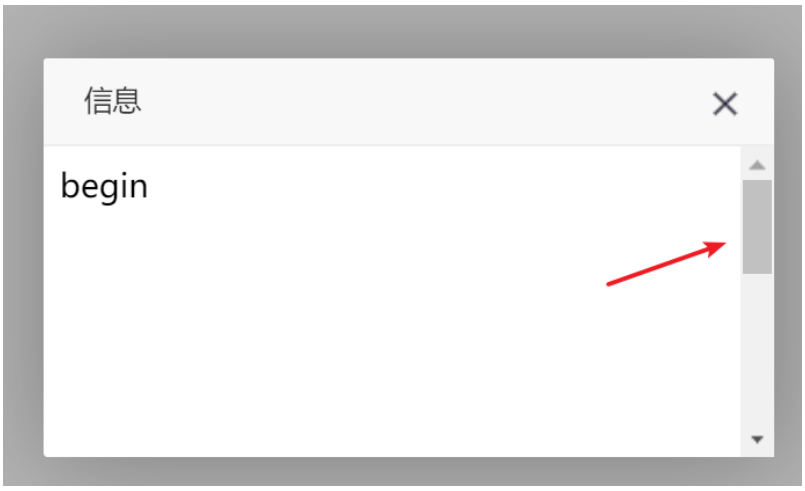
```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
    </head>
    <body>
        begin
        <br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br />
        end!
    </body>
</html>
```

运行文件代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            layui.use('layer', function() {
                var layer = layui.layer;

                var index = layer.open({
                    type: 2,
                    content: 'iframe.html',
                    success: function(layero, index) {}
                });
```

```
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



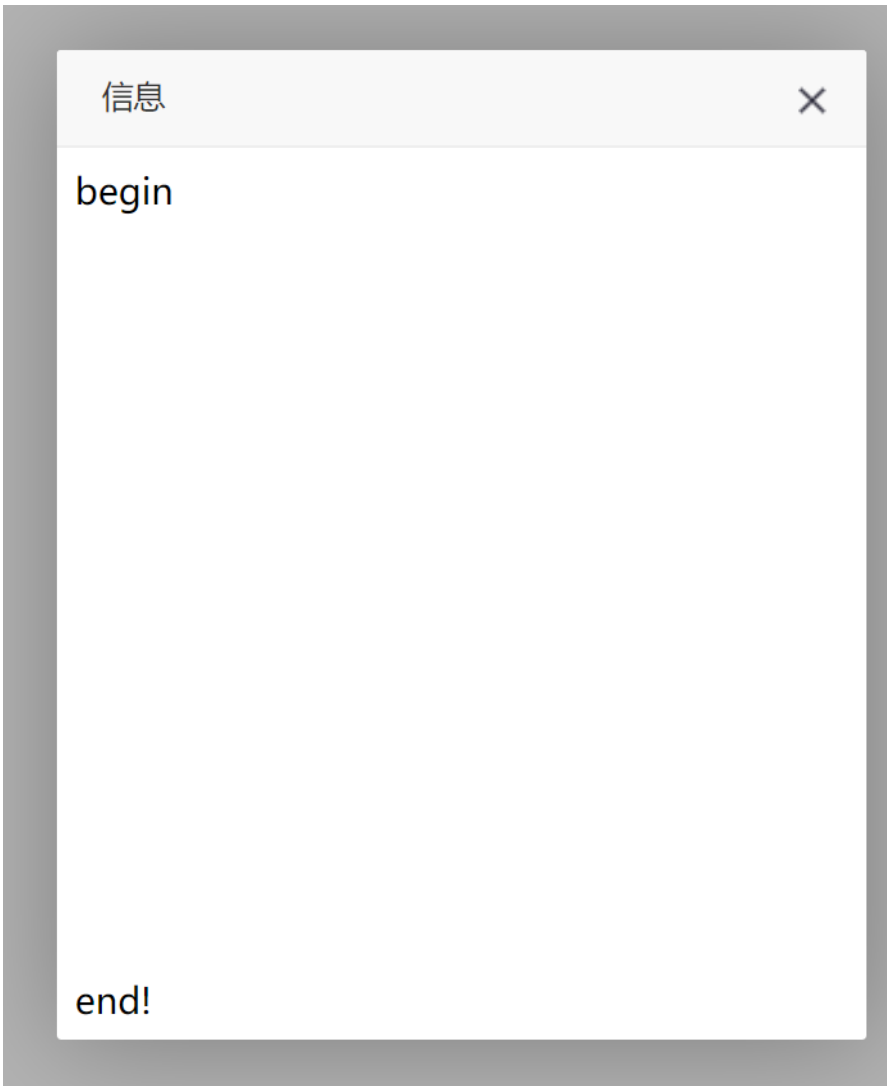
出现垂直滚动条，可以使用 iframeAuto(index)方法设置高度自适应。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            layui.use('layer', function() {
                var layer = layui.layer;

                var index = layer.open({
                    type: 2,
                    content: 'iframe.html',
                    success: function(layero, index) {
                        layer.iframeAuto(index);
                    }
                });
            });
        </script>
```

```
    </body>
</html>
```

运行后没有垂直滚动条，iframe 高度自适应了，如图 1-xx 所示。



1.18.50 layer.iframeSrc(index, url)重置特定 iframe url

使用方式：layer.iframeSrc(index, 'http://sentsin.com')

文件 iframe1.html 代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
    </head>
    <body>
```

```
        begin
        <br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br />
        end!
    </body>
</html>
```

文件 iframe2.html 代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
    </head>
    <body>
        newbegin
        <br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br />
        newend!
    </body>
</html>
```

运行文件代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;

            var index = layer.open({
                type: 2,
                content: 'iframe1.html',
                success: function(layero, index) {
                    layer.iframeAuto(index);
                }
            });
```

```
            setTimeout(function() {
                layer.iframeSrc(index, "iframe2.html");
            }, 3000);
        </script>
    </body>
</html>
```

1.18.51 layer.setTop(layero)置顶当前窗口

非常强大的一个方法，虽然一般很少用，但是当页面有很多很多 layer 窗口，需要像 Window 窗体那样点击某个窗口，该窗体就置顶在上面，那么 setTop 可以来轻松实现。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var layer = layui.layer;

            layer.open({
                type: 2,
                shade: false,
                area: '500px',
                content: 'http://www.layui.com',
                zIndex: layer.zIndex, //重点1
                success: function(layero) {
                    layer.setTop(layero); //重点2
                }
            });

            layer.open({
                type: 2,
                shade: false,
                area: '500px',
                content: 'http://www.layui.com',
                zIndex: layer.zIndex, //重点1
```

```
            success: function(layero) {
                layer.setTop(layero); //重点2
            }
        });

        layer.open({
            type: 2,
            shade: false,
            area: '500px',
            content: 'http://www.layui.com',
            zIndex: layer.zIndex, //重点1
            success: function(layero) {
                layer.setTop(layero); //重点2
            }
        });
    </script>
</body>
</html>
```

1.18.52 layer.full()、layer.min()、layer.restore()手工执行最大小化

文件 iframe.html 代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <input type="button" value="full" id="full"><br />
        <input type="button" value="min" id="min"><br />
        <input type="button" value="restore" id="restore"><br />

        <script src="layui/layui.all.js"></script>
        <script>
            $("#full").click(function() {
                var index = parent.layer.getFrameIndex(window.name);
                parent.layer.full(index);
            });

            $("#min").click(function() {
```

```
            var index = parent.layer.getFrameIndex(window.name);
            parent.layer.min(index);
        });

        $("#restore").click(function() {
            var index = parent.layer.getFrameIndex(window.name);
            parent.layer.restore(index);
        });
    </script>
</body>
</html>
```

运行文件代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var index = layer.open({
                type: 2,
                content: 'iframe.html',
                shade: false,
                maxmin: true
            });
        </script>
    </body>
</html>
```

1.18.53 layer.prompt(options, yes)输入层

options 不仅可支持传入基础参数，还可以传入 prompt 专用的属性。当然，也可以不传。yes 回调方法携带 value 表单值，index 索引，elem 表单元素。

prompt 层新定制的成员如下

```
{
  formType: 1, //输入框类型，支持 0（文本）默认 1（密码）2（多行文本）
  value: '', //初始时的值，默认空字符
  maxlength: 140, //可输入文本的最大长度，默认 500
```

```
    }
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <input type="button" value="button1" id="button1"><br />
        <input type="button" value="button2" id="button2"><br />
        <input type="button" value="button3" id="button3"><br />
        <input type="button" value="button4" id="button4"><br />
        <script src="layui/layui.all.js"></script>
        <script>
            $(document).ready(function() {
                $("#button1").click(function() {
                    layer.prompt(function(value, index, elem) {
                        alert(value); //得到value
                        layer.close(index);
                    });
                });
                $("#button2").click(function() {
                    layer.prompt({
                        formType: 0,
                        value: '初始值',
                        title: '请输入值',
                        area: ['800px', '350px'] //自定义文本域宽高
                    }, function(value, index, elem) {
                        alert(value); //得到value
                        layer.close(index);
                    });
                });
                $("#button3").click(function() {
                    layer.prompt({
                        formType: 1,
                        value: '初始值',
                        title: '请输入值',
                        area: ['800px', '350px'] //自定义文本域宽高
                    }, function(value, index, elem) {
                        alert(value); //得到value
```

```
                    layer.close(index);
                });
            });
            $("#button4").click(function() {
                layer.prompt({
                    formType: 2,
                    value: '初始值',
                    title: '请输入值',
                    area: ['800px', '350px'] //自定义文本域宽高
                }, function(value, index, elem) {
                    alert(value); //得到value
                    layer.close(index);
                });
            });
        });
    </script>
</body>
</html>
```

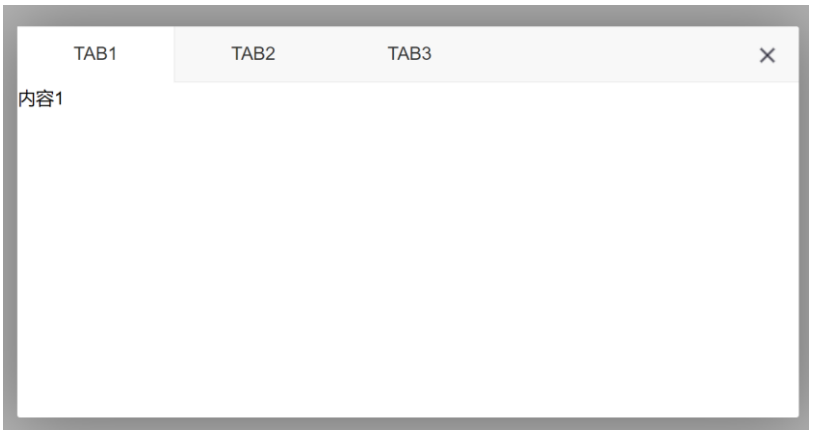
1.18.54 layer.tab(options)tab 层

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            layer.tab({
                area: ['600px', '300px'],
                tab: [{
                    title: 'TAB1',
                    content: '内容1'
                }, {
                    title: 'TAB2',
                    content: '内容2'
                }, {
                    title: 'TAB3',
```

```
                content: '内容3'
            }]
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.18.55 layer.photos(options)相册层

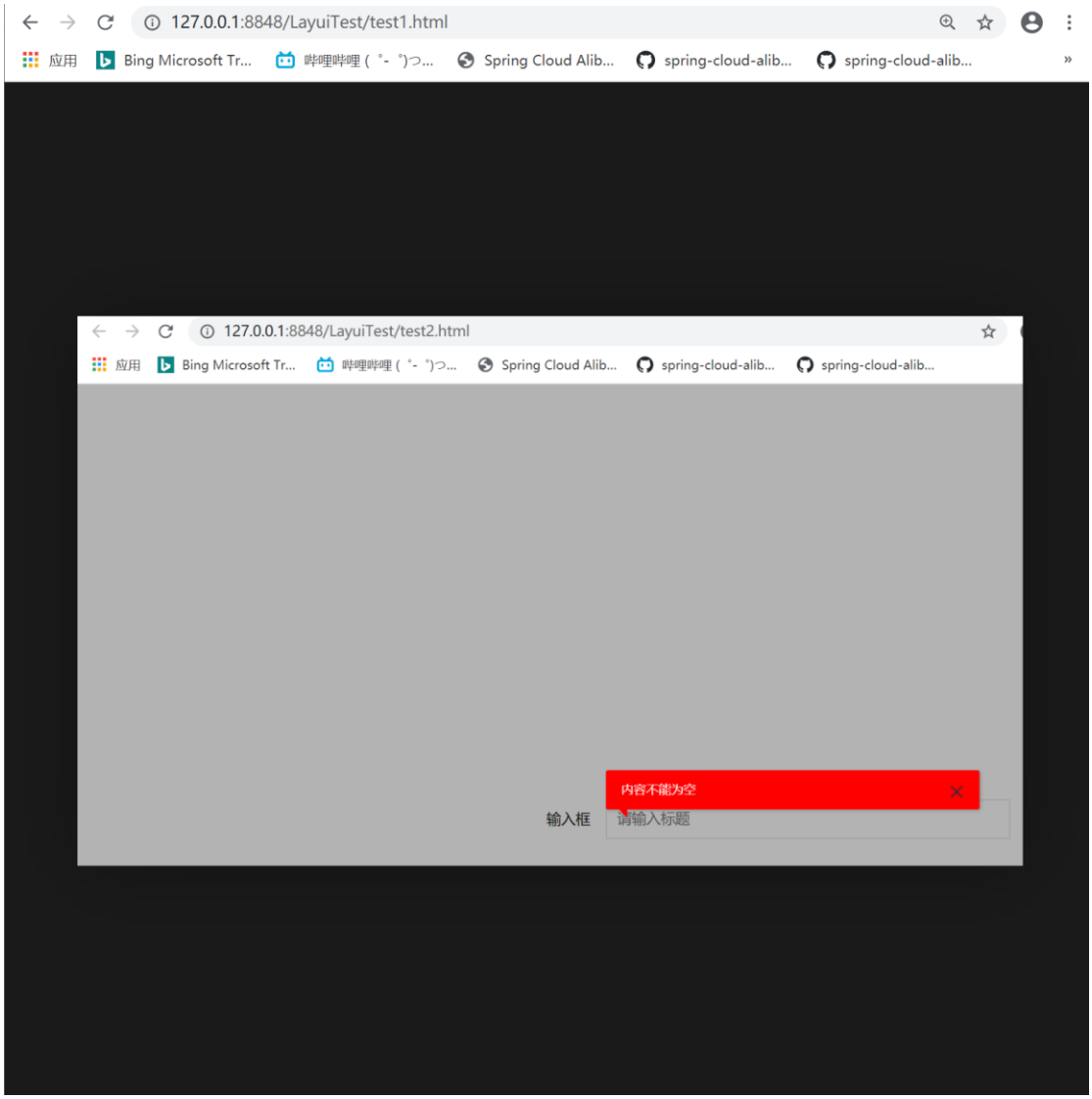
相册层，也可以称之为图片查看器。它的出场动画从 layer 内置的动画类型中随机展现。photos 支持传入 json 和直接读取页面图片两种方式。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            layer.photos({
                photos: {
                    "title": "", //相册标题
                    "id": 123, //相册id
                    "start": 1, //初始显示的图片序号，默认0
                    "data": [ //相册包含的图片，数组格式
```

```
                    {
                        "alt": "图片名",
                        "pid": 1, //图片id
                        "src": "4.png", //原图地址
                        "thumb": "" //缩略图地址
                    },
                    {
                        "alt": "图片名",
                        "pid": 2, //图片id
                        "src": "5.png", //原图地址
                        "thumb": "" //缩略图地址
                    }
                ]
            },
            anim: 5, //0-6的选择，指定弹出图片动画类型，默认随机。
            tab: function(pic, layero) {
                alert(pic); //当前图片的一些信息
            }
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



官方网址：
https://layer.layui.com/
中有若干常用使用案例，可以进行参考。

1.19 日期和时间组件 laydate

主要以年选择器、年月选择器、日期选择器、时间选择器、日期时间选择器，这五种类型的选择方式为基本核心使用方式，并且均支持范围选择（即双控件）。内置强劲的自定义日期格式解析和合法校正机制，含中文版和国际版，主题简约却又不失灵活多样。由于内部采用的是零依赖的原生 JavaScript 编写，因此又可作为独立组件使用。

模块加载名称：laydate。

独立版本官方网址：

http://www.layui.com/laydate

1.19.1 使用方式

laydate 有两种使用方式，如图 1-xx 所示。

和 layer 一样，你可以在 layui 中使用 layDate，也可直接使用 layDate 独立版，请按照你的实际需求来选择。

场景	用前准备	调用方式
1. 在 layui 模块中使用	下载 layui 后，引入 *layui.css* 和 *layui.js* 即可	通过 *layui.use('laydate', callback)* 加载模块后，再调用方法
2. 作为独立组件使用	去 layDate 独立版本官网下载组件包，引入 laydate.js 即可	直接调用方法使用

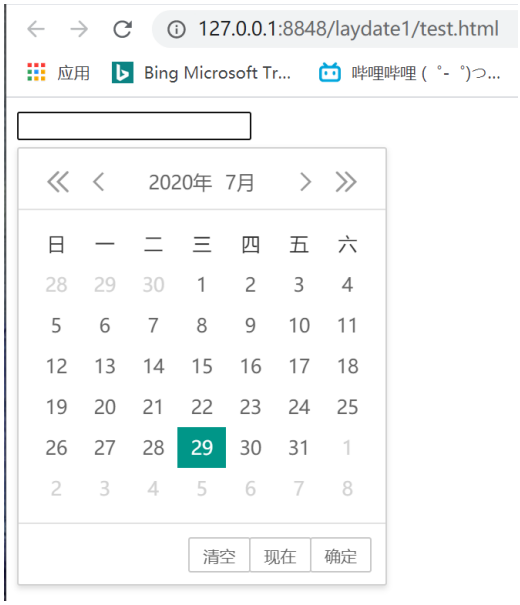
1.19.1.1 独立使用

当独立使用时，需要到官网：
http://www.layui.com/laydate/
下载 laydate，并在项目中进行引用，官网提供了 demo.html 做为使用案例。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="laydate/laydate.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <!-- 注意：这一层元素并不是必须的 -->
            <input type="text" class="layui-input" id="test1">
        </div>
        <script>
            //执行一个laydate实例
            laydate.render({
                elem: '#test1' //指定元素
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

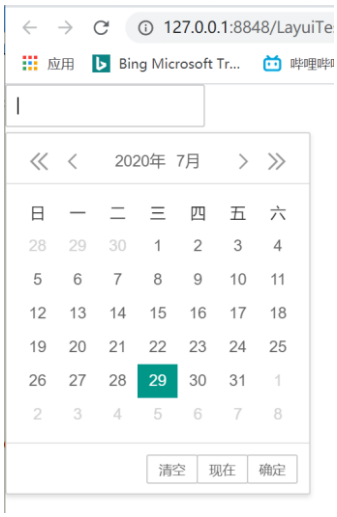


1.19.1.2 模块化使用

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <!-- 注意：这一层元素并不是必须的 -->
            <input type="text" class="layui-input" id="test1">
        </div>
        <script>
            var laydate = layui.laydate;
            //执行一个laydate实例
            laydate.render({
                elem: '#test1' //指定元素
            });
        </script>

    </body>
</html>
```

运行效果如图 1-xx 所示。



1.19.2 基础参数选项

通过核心方法：
laydate.render(options);
来设置基础参数。

也可以通过方法：
laydate.set(options);
来设置全局基础参数。

1.19.3 elem 绑定元素

类型：String/DOM，默认值：无
必填项，用于绑定执行日期渲染的元素，值一般为选择器或 DOM 对象。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
```

```
            <input type="text" class="layui-input" id="test1">
            <br />
            <input type="text" class="layui-input" id="test2">
        </div>
        <script>
            var laydate = layui.laydate;
            laydate.render({
                elem: '#test1'
            });
            laydate.render({
                elem: document.getElementById('test2')
            });
        </script>
    </body>
</html>
```

1.19.4 type 控件选择类型

类型：String，默认值 date。

用于单独提供不同的选择器类型，可选值如图 1-xx 所示。

type可选值	名称	用途
year	年选择器	只提供年列表选择
month	年月选择器	只提供年、月选择
date	日期选择器	可选择：年、月、日。type默认值，一般可不填
time	时间选择器	只提供时、分、秒选择
datetime	日期时间选择器	可选择：年、月、日、时、分、秒

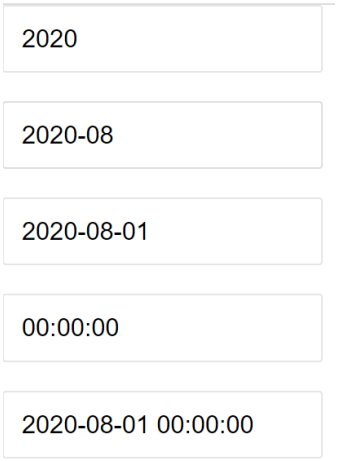
测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
```

```
<div class="layui-inline">
    <input type="text" class="layui-input" id="test1">
    <br />
    <input type="text" class="layui-input" id="test2">
    <br />
    <input type="text" class="layui-input" id="test3">
    <br />
    <input type="text" class="layui-input" id="test4">
    <br />
    <input type="text" class="layui-input" id="test5">
</div>

<script>
    var laydate = layui.laydate;

    //年选择器
    laydate.render({
        elem: '#test1',
        type: 'year'
    });

    //年月选择器
    laydate.render({
        elem: '#test2',
        type: 'month'
    });

    //日期选择器
    laydate.render({
        elem: '#test3'
        //,type: 'date' //默认，可不填
    });

    //时间选择器
    laydate.render({
        elem: '#test4',
        type: 'time'
    });

    //日期时间选择器
    laydate.render({
        elem: '#test5',
        type: 'datetime'
    });
```

```
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.19.5 range 开启左右面板范围选择

类型：Boolean/String，默认值 false。

如果设置 true，将默认采用“-”分割。 也可以直接设置分割字符。五种选择器类型均支持左右面板的范围选择。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input" id="test1">
            <br />
            <input type="text" class="layui-input" id="test2">
            <br />
            <input type="text" class="layui-input" id="test3">
            <br />
            <input type="text" class="layui-input" id="test4">
            <br />
```

```
            <input type="text" class="layui-input" id="test5">
        </div>
        <script>
            var laydate = layui.laydate;

            //年范围选择
            laydate.render({
                elem: '#test1',
                type: 'year',
                range: true //或 range: '~' 来自定义分割字符
            });

            //年月范围选择
            laydate.render({
                elem: '#test2',
                type: 'month',
                range: '~' //或 range: '~' 来自定义分割字符
            });

            //日期范围选择
            laydate.render({
                elem: '#test3',
                range: '~' //或 range: '~' 来自定义分割字符
            });

            //时间范围选择
            laydate.render({
                elem: '#test4',
                type: 'time',
                range: '~' //或 range: '~' 来自定义分割字符
            });

            //日期时间范围选择
            laydate.render({
                elem: '#test5',
                type: 'datetime',
                range: '~' //或 range: '~' 来自定义分割字符
            });
        </script>
    </body>
</html>
```

1.19.6 format 自定义格式

类型：String，默认值 yyyy-MM-dd。

通过日期时间各自的格式符和长度来设置一个所需要的日期格式。layDate 支持的格式如图 1-xx 所示。

格式符	说明
yyyy	年份，至少四位数。如果不足四位，则前面补零
y	年份，不限制位数，即不管年份多少位，前面均不补零
MM	月份，至少两位数。如果不足两位，则前面补零。
M	月份，允许一位数。
dd	日期，至少两位数。如果不足两位，则前面补零。
d	日期，允许一位数。
HH	小时，至少两位数。如果不足两位，则前面补零。
H	小时，允许一位数。
mm	分钟，至少两位数。如果不足两位，则前面补零。
m	分钟，允许一位数。
ss	秒数，至少两位数。如果不足两位，则前面补零。
s	秒数，允许一位数。

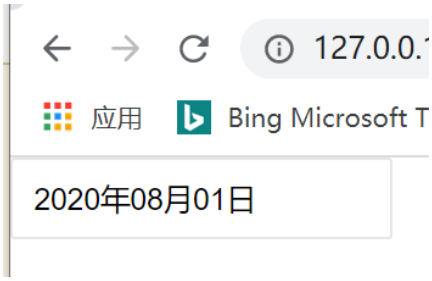
通过上述不同的格式符组合成一段日期时间字符串，可任意排版，如图 1-xx 所示。

格式	示例值
yyyy-MM-dd HH:mm:ss	2017-08-18 20:08:08
yyyy年MM月dd日 HH时mm分ss秒	2017年08月18日 20时08分08秒
yyyyMMdd	20170818
dd/MM/yyyy	18/08/2017
yyyy年M月	2017年8月
M月d日	8月18日
北京时间：HH点mm分	北京时间：20点08分
yyyy年的M月某天晚上，大概H点	2017年的8月某天晚上，大概20点

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input" id="test1">
        </div>

        <script>
            var laydate = layui.laydate;
            //自定义日期格式
            laydate.render({
                elem: '#test1',
                format: 'yyyy年MM月dd日' //可任意组合
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.19.7 value 初始值

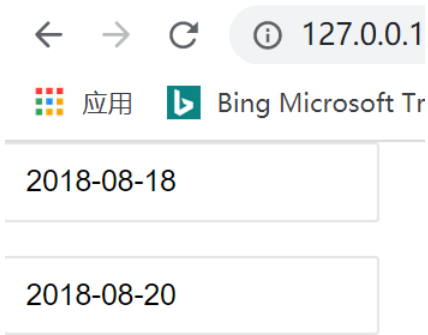
类型：String，默认值 new Date()。
支持传入符合 format 参数设置的日期格式字符，或者 new Date()。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
```

```
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input" id="test1">
            <br />
            <input type="text" class="layui-input" id="test2">
        </div>

        <script>
            var laydate = layui.laydate;

            //传入符合format格式的字符给初始值
            laydate.render({
                elem: '#test1',
                value: '2018-08-18' //必须遵循format参数设置的格式
            });

            //传入Date对象给初始值
            laydate.render({
                elem: '#test2',
                value: new Date(1534766888000) //参数即为：2018-08-20
20:08:08 的时间戳
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



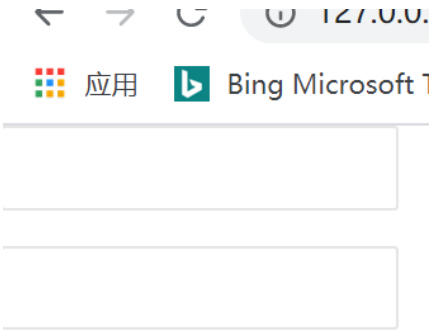
1.19.8 isInitValue 初始值填充

类型：Boolean，默认值 true。
用于控制是否自动向元素填充初始值（需配合 value 参数使用）。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input" id="test1">
            <br />
            <input type="text" class="layui-input" id="test2">
        </div>

        <script>
            var laydate = layui.laydate;

            //传入符合format格式的字符给初始值
            laydate.render({
                elem: '#test1',
                value: '2018-08-18', //必须遵循format参数设置的格式
                isInitValue: false
            });

            //传入Date对象给初始值
            laydate.render({
                elem: '#test2',
                value: new Date(1534766888000), //参数即为：2018-08-20
20:08:08 的时间戳
                isInitValue: false
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.19.9 min/max 最小/最大范围内的日期时间值

类型：string，默认值 min: '1900-1-1'、max: '2099-12-31'。

设置有限范围内的日期或时间值，不在范围内的将不可选中。这两个参数的赋值非常灵活，主要有以下几种情况：

1.	如果值为字符类型，则：*年月日必须用-（中划线）分割、时分秒必须用：（半角冒号）号分割*。这里并非遵循 format 设定的格式
2.	如果值为整数类型，且数字＜86400000，则数字代表天数，如：min: -7，即代表最小日期在7天前，正数代表若干天后
3.	如果值为整数类型，且数字≥86400000，则数字代表时间戳，如：max: 4073558400000，即代表最大日期在：公元3000年1月1日

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input" id="test1">
            <br />
            <input type="text" class="layui-input" id="test2">
            <br />
            <input type="text" class="layui-input" id="test3">
            <br />
            <input type="text" class="layui-input" id="test4">
        </div>

        <script>
```

```
            var laydate = layui.laydate;

            //日期有效范围只限定在：2017年
            laydate.render({
                elem: '#test1',
                min: '2017-1-1',
                max: '2017-12-31'
            });

            //日期有效范围限定在：过去一周到未来一周
            laydate.render({
                elem: '#test2',
                min: -7, //7天前
                max: 7 //7天后
            });

            //日期时间有效范围的设置：
            laydate.render({
                elem: '#test3',
                type: 'datetime',
                min: '2017-8-11 12:30:00',
                max: '2017-8-18 12:30:00'
            });

            //时间有效范围设置在：上午九点半到下午五点半
            laydate.render({
                elem: '#test4',
                type: 'time',
                min: '09:30:00',
                max: '17:30:00'
            });
        </script>
    </body>
</html>
```

毫不保留地说，min 和 max 参数是两个非常强大的存在，合理运用，可帮助用户在日期与时间的选择上带来更为友好的约束与体验。

1.19.10 trigger 自定义弹出控件的事件

类型：String，默认值 focus。如果绑定的元素非输入框，则默认事件为：click。

测试代码如下：

```
<!DOCTYPE html>
```

```
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input">
            <br />
            <input type="text" class="layui-input" id="test1">
            <br />
            <input type="text" class="layui-input">
            <br />
            <input type="text" class="layui-input" id="test2">
            <br />
            <span id="test3">2008-08-08</span>
        </div>
        <script>
            var laydate = layui.laydate;
            laydate.render({
                elem: '#test1',
            });
            laydate.render({
                elem: '#test2',
                trigger: 'click' //采用click弹出
            });
            laydate.render({
                elem: '#test3',
                trigger: 'click' //采用click弹出
            });
        </script>
    </body>
</html>
```

1.19.11 show 默认显示

类型：Boolean，默认值 false。
如果设置: true，则控件默认显示在绑定元素的区域。通常用于外部事件调用控件。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input" id="test1">
            <input type="text" class="layui-input" id="test2">
            <input type="text" class="layui-input" id="test3">
        </div>

        <script>
            var laydate = layui.laydate;
            //默认显示
            laydate.render({
                elem: '#test1',
                show: true //直接显示
            });

            //外部事件调用
            lay('#test2').on('click', function(e) { //假设test2是一个按钮
                laydate.render({
                    elem: '#test3',
                    show: true, //直接显示
                    closeStop: '#test2',
                    //这里代表的意思是：点击test2所在元素阻止关闭事件冒泡。
                    //如果不设置，则无法弹出控件
                });
            });
        </script>
    </body>
</html>
```

1.19.12 position 定位方式

类型：String，默认值 absolute。
用于设置控件的定位方式，有如图 1-xx 所示的三种可选值。

position 可选值	说明
abolute	绝对定位，始终吸附在绑定元素周围。默认值
fixed	固定定位，初始吸附在绑定元素周围，不随浏览器滚动条所左右。*一般用于在固定定位的弹层中使用。*
static	静态定位，控件将直接嵌套在指定容器中。 注意：*请勿与 show 参数的概念搞混淆。show为 true 时，控件仍然是采用绝对或固定定位。而这里是直接嵌套显示*

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input" id="test1">
            <br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br />
            <input type="text" class="layui-input" id="test2">
        </div>
        <br />
        <br />
        <br />
        <div class="layui-inline">
            <div id="test3"></div>
        </div>
        <span id="testView"></span>
        <script>
            var laydate = layui.laydate;
            //abolute绝对定位，始终吸附在绑定元素周围，是默认值。
            laydate.render({
                elem: '#test1',
                position: 'abolute',
                change: function(value, date) { //监听日期被切换
                    $('#testView').html(value);
                }
            });
            //fixed固定定位，初始吸附在绑定元素周围，
```

```
            //不随浏览器滚动条所左右。一般用于在固定定位的弹层中使用。
            laydate.render({
                elem: '#test2',
                position: 'fixed',
                change: function(value, date) { //监听日期被切换
                    $('#testView').html(value);
                }
            });
            //static嵌套在指定容器中。
            laydate.render({
                elem: '#test3',
                position: 'static',
                change: function(value, date) { //监听日期被切换
                    $('#testView').html(value);
                }
            });
        </script>
    </body>
</html>
```

1.19.13 zIndex 层叠顺序

类型：Number，默认值 66666666。

一般用于解决与其它元素的互相被遮掩的问题。如果 position 参数设为 static 时，该参数无效。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input" id="test1">
        </div>
        <div style="width: 1000px;height:
1000px;z-index:66666667;background-color: red;position: absolute;">
        </div>
```

```
        <script>
            var laydate = layui.laydate;
            //设置控件的层叠顺序
            laydate.render({
                elem: '#test1',
                zIndex: 66666666
            });
        </script>
    </body>
</html>
```

被覆盖了。

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input" id="test1">
        </div>
        <div style="width: 1000px;height:
1000px;z-index:66666667;background-color: red;position: absolute;">
        </div>
        <script>
            var laydate = layui.laydate;
            //设置控件的层叠顺序
            laydate.render({
                elem: '#test1',
                zIndex: 66666668
            });
        </script>
    </body>
</html>
```

正确显示日期弹出框。

1.19.14 showBottom 是否显示底部栏

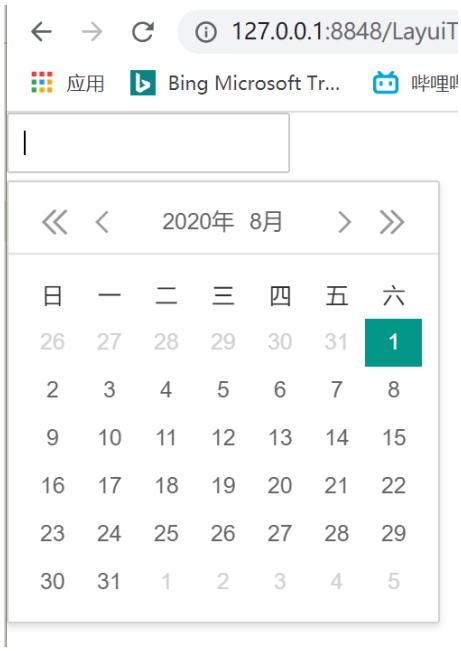
类型：Boolean，默认值 true。

如果设置 false，将不会显示控件的底部栏区域，底部栏区域如图 1-xx 所示。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input" id="test1">
        </div>

        <script>
            var laydate = layui.laydate;
            //不显示底部栏
            laydate.render({
                elem: '#test1',
                showBottom: false
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.19.15 btns 工具按钮

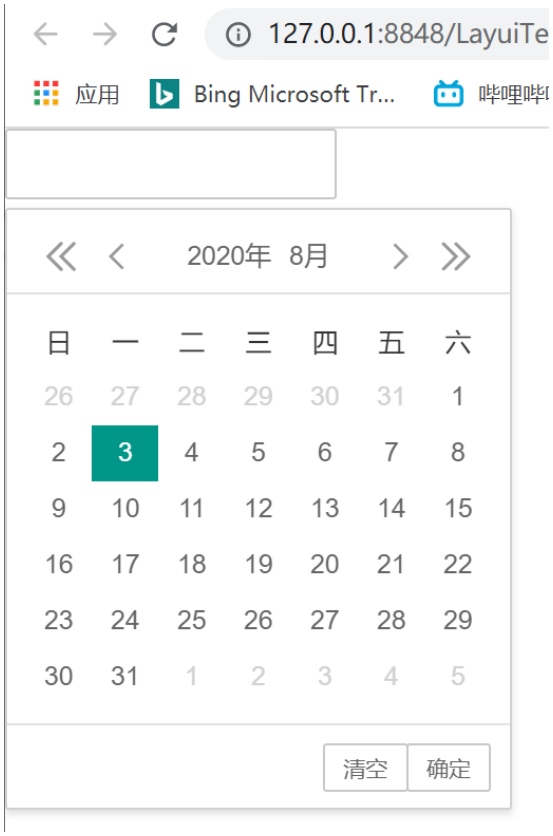
类型：Array，默认值['clear', 'now', 'confirm']。
右下角显示的按钮，会按照数组顺序排列，内置可识别的值有：clear、now、confirm。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input" id="test1">
        </div>

        <script>
            var laydate = layui.laydate;
            //只显示清空和确认
            laydate.render({
                elem: '#test1',
                btns: ['clear', 'confirm']
```

```
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.19.16 lang 语言

类型：String，默认值 cn。

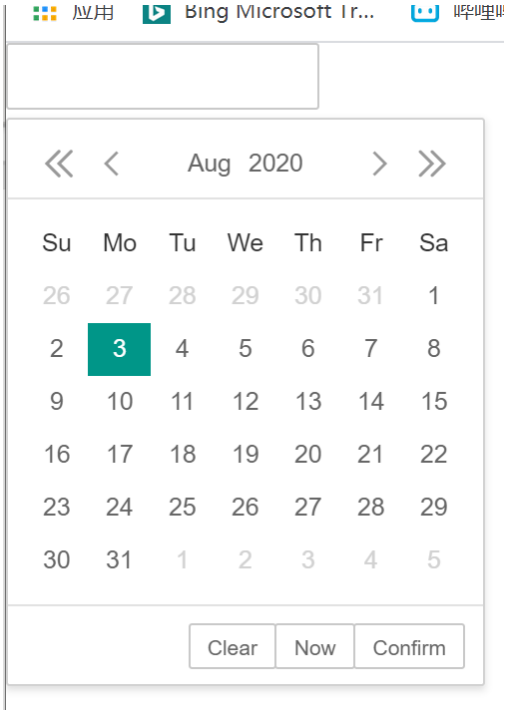
内置了两种语言版本：cn（中文版）、en（国际版，即英文版）。没有开放自定义语言，是为了避免繁琐的配置。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
```

```
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input" id="test1">
        </div>

        <script>
            var laydate = layui.laydate;
            //国际版
            laydate.render({
                elem: '#test1',
                lang: 'en'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.19.17 theme 主题

类型：String，默认值：default

内置了多种主题，theme 的可选值有：default（默认简约）、molv（墨绿背景）、#颜色值（自定义颜色背景）、grid（格子主题）

测试代码如下：

```
<!DOCTYPE html>
```

```
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input" id="test1">
            <input type="text" class="layui-input" id="test2">
            <input type="text" class="layui-input" id="test3">
        </div>

        <script>
            var laydate = layui.laydate;
            //墨绿背景主题
            laydate.render({
                elem: '#test1',
                theme: 'molv'
            });

            //自定义背景色主题 - 非常实用
            laydate.render({
                elem: '#test2',
                theme: '#393D49'
            });

            //格子主题
            laydate.render({
                elem: '#test3',
                theme: 'grid'
            });
        </script>
    </body>
</html>
```

另外，还可以传入其它字符，比如：
theme: 'xxx'
那么控件将会多出一个 class="laydate-theme-xxx"的 CSS 类，以便单独定制主题。

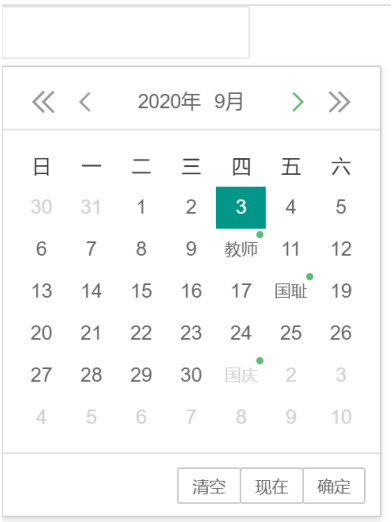
1.19.18 calendar 是否显示公历节日

类型：Boolean，默认值 false。
内置了一些我国通用的公历重要节日，通过设置 true 来开启。国际版不会显示。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input" id="test1">
        </div>

        <script>
            var laydate = layui.laydate;
            //允许显示公历节日
            laydate.render({
                elem: '#test1',
                calendar: true
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.19.19 mark 标注重要日子

类型：Object，默认值无。
可以自定义标注重要日子，比如结婚纪念日，日程等等。它分为以下三种：

标注	格式	说明
每年的日期	{'0-9-18': '国耻'}	0 即代表每一年
每月的日期	{'0-0-15': '中旬'}	0-0 即代表每年每月（layui 2.1.1/layDate 5.0.4 新增）
特定的日期	{'2017-8-21': '发布'}	-

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input" id="test1">
        </div>

        <script>
            var laydate = layui.laydate;
            //标注重要日子
```

```
            var ins1 = laydate.render({
                elem: '#test1',
                mark: {
                    '0-03-06': '生日',
                    '0-12-31': '跨年', //每年12月31日
                    '0-0-10': '工资', //每个月10号
                    '2017-8-15': '', //具体日期
                    '2020-8-3': '学习', //如果为空字符，则默认显示数字+徽章
                    '2020-8-21': '完毕'
                },
                done: function(value, date) {
                    if (date.year === 2017 && date.month === 8 && date.date
=== 15) { //点击2017年8月15日，弹出提示语
                        ins1.hint('中国人民抗日战争胜利72周年');
                    }
                }
            });
        </script>
    </body>
</html>
```

1.19.20 ready 控件初始打开的回调

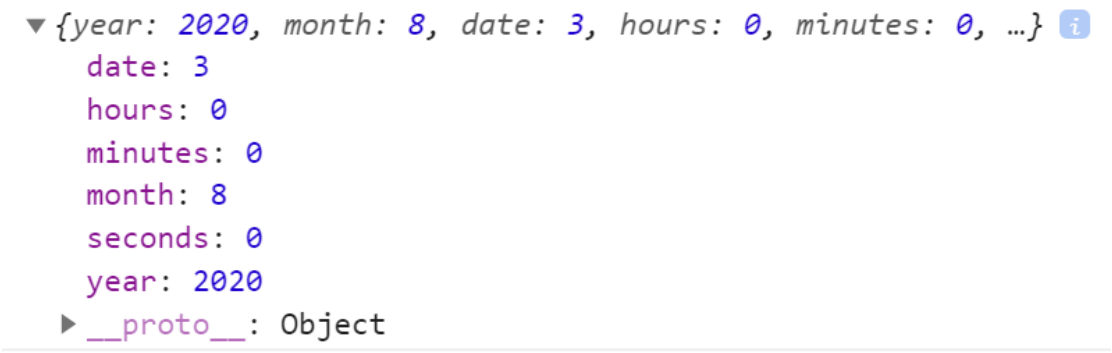
控件在打开时触发，回调返回一个参数：初始的日期时间对象。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input" id="test1">
        </div>

        <script>
            var laydate = layui.laydate;
            laydate.render({
                elem: '#test1',
```

```
            ready: function(date) {
                console.log(date); //得到初始的日期时间对象：{year: 2017, month: 8, date: 18, hours: 0, minutes: 0, seconds: 0}
            }
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.19.21 done 控件选择完毕后的回调

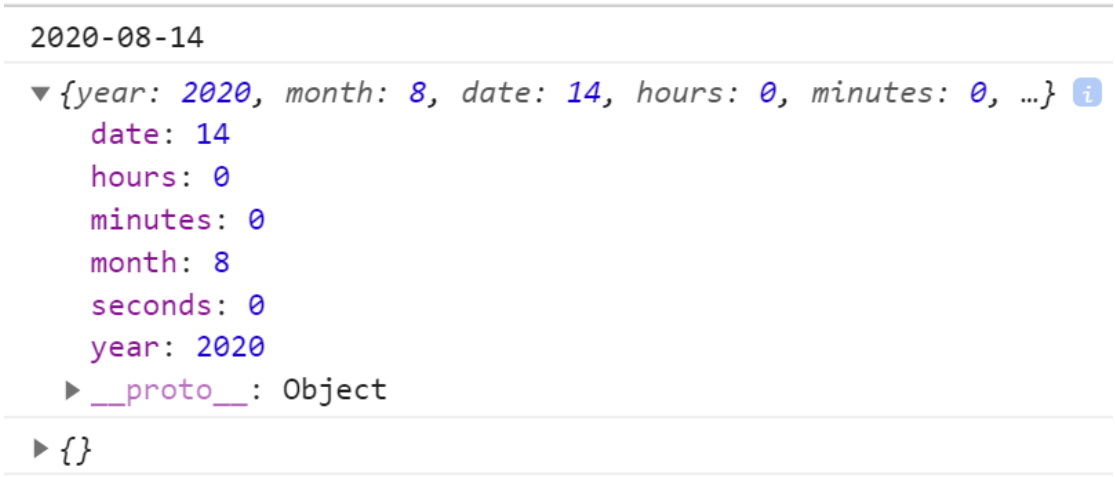
点击"日期"、"清空"、"现在"、"确定"均会触发。回调返回三个参数：生成的值、日期时间对象、结束的日期时间对象

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input" id="test1">
        </div>
        <script>
            var laydate = layui.laydate;

            laydate.render({
```

```
                elem: '#test1',
                done: function(value, date, endDate) {
                    console.log(value);
                    //得到日期生成的值，如：2017-08-18
                    console.log(date);
                    //得到日期时间对象：{year: 2017, month: 8, date: 18, hours: 0, minutes: 0, seconds: 0}
                    console.log(endDate);
                    //得结束的日期时间对象，开启范围选择（range: true）才会返回。对象成员同上。
                }
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.19.22 其它方法

其它方法如图 1-xx 所示。

方法名	备注

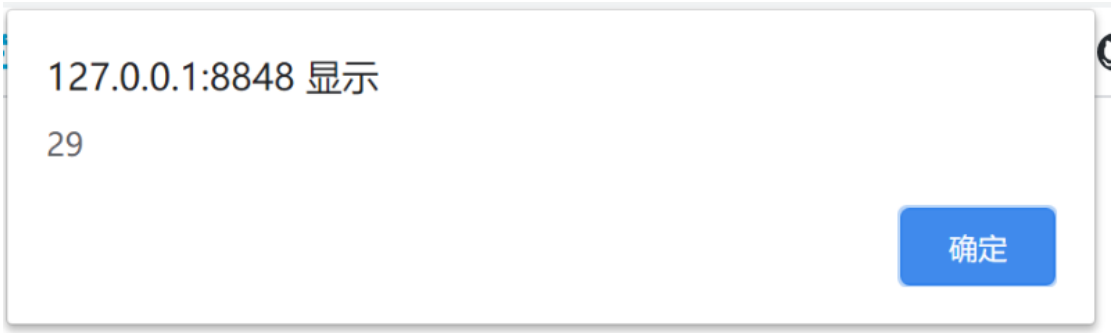
laydate.getEndDate(month, year)	获取指定年月的最后一天，month默认为当前月，year默认为当前年。如：
	var endDate1 = laydate.getEndDate(10); //得到31
	var endDate2 = laydate.getEndDate(2, 2080); //得到29

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
```

```
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-inline">
            <input type="text" class="layui-input" id="test1">
        </div>
        <script>
            var laydate = layui.laydate;
            alert(laydate.getEndDate(2, 2020));
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.20 分页组件 laypage

layPage 致力于提供极致的分页逻辑，既可轻松胜任异步分页，也可作为页面刷新式分页。

模块加载名称：laypage。

1.20.1 快速使用

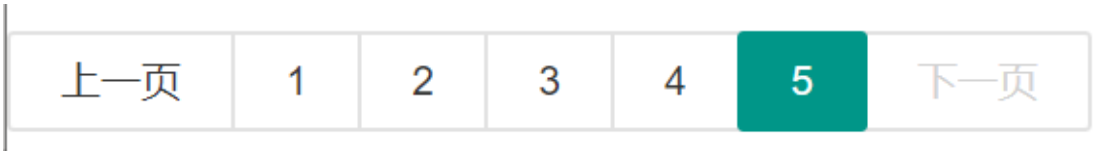
laypage 的使用非常简单，指向一个用于存放分页的容器，通过服务端得到一些初始值，即可完成分页渲染

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
```

```
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var laypage = layui.laypage;
            //执行一个laypage实例
            laypage.render({
                elem: 'test1',
                //注意，这里的 test1 是 ID，不用加 # 号
                count: 50 //数据总数，从服务端得到
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.20.2 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如图 1-xx 所示 。

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
	01. //开启location.hash的记录
	02. laypage.render({
	03. elem: 'test1'
	04. ,count: 500
	05. ,curr: location.hash.replace('#!fenye=', '') //获取起始页
	06. ,hash: 'fenye' //自定义hash值
	07. });
	08.
		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

1.20.2.1 elem: document.getElementById("test1")

elem 值类型 String/Object。指向存放分页的容器，值可以是容器 ID、DOM 对象，例如：

（1）elem: 'id' 。注意：这里不能加#号。

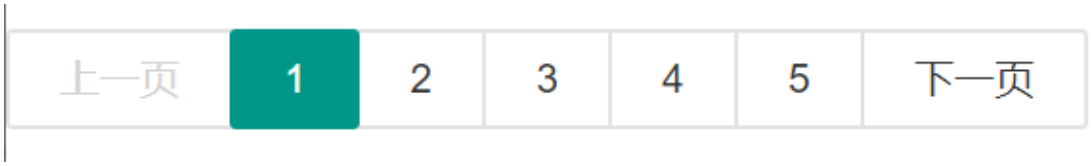
（2）elem: document.getElementById('id')。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
```

```
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var laypage = layui.laypage;
            //执行一个laypage实例
            laypage.render({
                elem: document.getElementById("test1"),
                count: 50 //数据总数，从服务端得到
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.20.2.2 count

count 值类型 Number。
数据总记录数，一般通过服务端得到。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var laypage = layui.laypage;
```

```
            //执行一个laypage实例
            laypage.render({
                elem: "test1",
                count: 500 //数据总数，从服务端得到
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.20.2.3 limit

limit 值类型 Number。
每页显示的条数。laypage 将会借助 count 和 limit 计算出分页数。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var laypage = layui.laypage;
            //执行一个laypage实例
            laypage.render({
                elem: "test1",
                count: 500, //数据总数，从服务端得到
                limit: 200
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.20.2.4 layout

layout 值类型 Array。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）、skip（快捷跳页区域）。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var laypage = layui.laypage;
            laypage.render({
                elem: "test1",
                count: 5000,
                layout: ['prev', 'page', 'next', 'limit', 'refresh', 'skip',
'count']
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

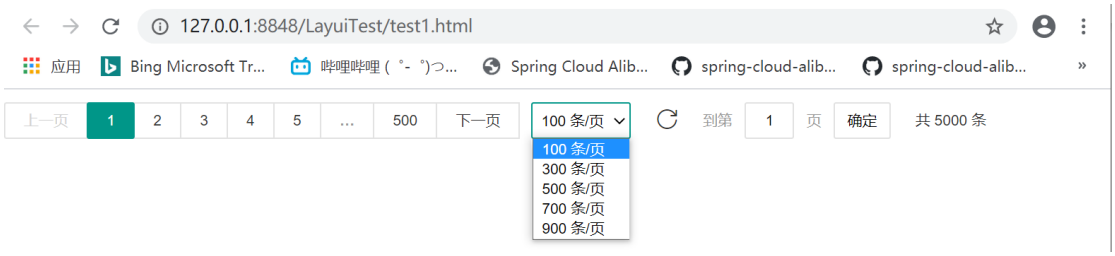
1.20.2.5 limits

limits 值类型 Array。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。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var laypage = layui.laypage;
            laypage.render({
                elem: "test1",
                count: 5000,
                limits: [100, 300, 500, 700, 900],
                layout: ['prev', 'page', 'next', 'limit', 'refresh', 'skip',
'count']
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



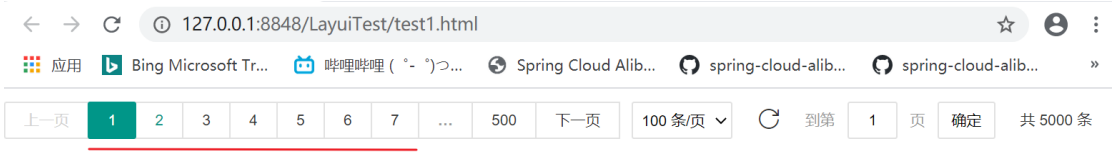
1.20.2.6 groups

groups 值类型 Number。连续出现的页码个数。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var laypage = layui.laypage;
            laypage.render({
                elem: "test1",
                count: 5000,
                groups: 7,
                limits: [100, 300, 500, 700, 900],
                layout: ['prev', 'page', 'next', 'limit', 'refresh', 'skip',
'count']
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.20.2.7 first-prev-next-last

first 值类型 String。自定义“首页”的内容，支持传入普通文本和 HTML。
prev 值类型 String。自定义“上一页”的内容，支持传入普通文本和 HTML。
next 值类型 String。自定义“下一页”的内容，支持传入普通文本和 HTML。
last 值类型 String。自定义“尾页”的内容，支持传入普通文本和 HTML。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <div id="test2"></div>
        <script>
            var laypage = layui.laypage;
            laypage.render({
                elem: "test1",
                count: 5000,
                first: "我是首页",
                prev: "我是上一页",
                next: "我是下一页",
                last: "我是末页",
                layout: ['prev', 'page', 'next', 'limit', 'refresh', 'skip',
'count']
            });

            laypage.render({
                elem: "test2",
                count: 5000,
                first: "我是首页",
                prev: "我是上一页",
                next: "我是下一页",
                last: "我是末页"
            });
        </script>
```

```
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.20.2.8 theme

theme 值类型 String。自定义主题。支持传入：颜色值或任意普通字符，例如：

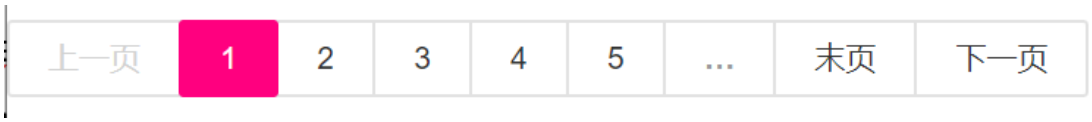
（1）theme: '#c00'。

（2）theme: 'xxx'。将会生成 class="layui-laypage-xxx"的 CSS 类，以便自定义主题。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var laypage = layui.laypage;
            laypage.render({
                elem: "test1",
                theme: "#ff007f",
                count: 5000,
                first: "首页",
                prev: "上一页",
                next: "下一页",
                last: "末页"
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.20.2.9 curr

curr 值类型 Number。起始页。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var laypage = layui.laypage;
            laypage.render({
                elem: "test1",
                theme: "#ff007f",
                curr: 88,
                count: 5000,
                first: "首页",
                prev: "上一页",
                next: "下一页",
                last: "末页"
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.20.2.10 hash

当使用如下代码创建分页：

```
<div id="test1"></div>
<script>
    var laypage = layui.laypage;
    laypage.render({
        elem: "test1",
        theme: "#ff007f",
        curr: 88,
        count: 5000,
        first: "首页",
        prev: "上一页",
        next: "下一页",
        last: "末页"
    });
</script>
```

打开网页时自动定位到 88 页，使用鼠标点击 87 页，把 URL 发给其它人，其它人看到的页数还是 88 页，这是不对的，其它人应该看到 87 页才是正确的，所以要使用 hash 属性。

hash 值类型 String/Boolean。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}，利用这个参数可以在页面载入时就定位到指定页。

一般用于刷新类型的跳页以及 HASH 跳页，例如：

```
//开启 location.hash 的记录
laypage.render({
  elem: 'test1',
  count: 500,
  curr: location.hash.replace('#!fenye=', ''), //获取起始页
  hash: 'fenye' //自定义 hash 值
});
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
```

```
        <script>
            var laypage = layui.laypage;
            laypage.render({
                elem: "test1",
                theme: "#ff007f",
                count: 5000,
                curr: location.hash.replace('#!gotoPage=', ''),
                hash: "gotoPage",
                first: "首页",
                prev: "上一页",
                next: "下一页",
                last: "末页"
            });
        </script>
    </body>
</html>
```

把 URL 给别人，别人看到的指定页数是一样的。

1.20.3 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）

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var laypage = layui.laypage;

            laypage.render({
                elem: 'test1',
                count: 70, //数据总数，从服务端得到
                jump: function(obj, first) {
                    //obj包含了当前分页的所有参数，比如：
```

```
                alert(obj.curr); //得到当前页，以便向服务端请求对应页的数
据。
                alert(obj.limit); //得到每页显示的条数
                //首次不执行
                if (!first) {} else {
                    alert("first run begin query!");
                }
            }
        });
    </script>
</body>
</html>
```

1.20.4 laypage 实现服务端分页公共代码

laypage 实现服务端分页公共代码。

1.20.4.1 工具类代码

```
package dbtools;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class GetConnection {

    public static Connection getConnection() {
        Connection conn = null;
        try {
            String url =
"jdbc:mysql://localhost:3306/y2?serverTimezone=Asia/Shanghai";
            String username = "root";
            String password = "123123";
            String driverName = "com.mysql.jdbc.Driver";
            Class.forName(driverName);
            conn = DriverManager.getConnection(url, username,
password);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return conn;
```

```
    }
}
```

1.20.4.2 DAO 类代码

```
package dao;

import dbtools.GetConnection;
import dto.PageDTO;
import dto.UserinfoPageQueryDTO;
import entity.Userinfo;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

public class UserinfoDAO {
    public PageDTO selectUserinfoPage(UserinfoPageQueryDTO dto) {
        PageDTO pageDTO = new PageDTO();
        List<Userinfo> listUserinfo = new ArrayList<>();
        try {
            int pageSize = 12;
            int beginPosition = (dto.getGotoPage() - 1) * pageSize;

            String sql1 = "select count(*) recordCount from userinfo";
            String sql2 = "select * from userinfo";
            String where = " where 1=1";
            String order = " order by id asc";
            String limit = " limit " + beginPosition + "," + pageSize;

            dto.getUsername();
            dto.getPassword();
            dto.getBeginAge();
            dto.getEndAge();
            dto.getBeginInsertDate();
            dto.getEndInsertDate();

            List sqlParamValueList = new ArrayList();
            if (!"".equals(dto.getUsername()) && dto.getUsername() !=
```

```
null) {
                where = where + " and username like ?";
                sqlParamValueList.add("%" + dto.getUsername() + "%");
                pageDTO.getQueryParam().put("username",
dto.getUsername());
            }
            if (!"".equals(dto.getPassword()) && dto.getPassword() !=
null) {
                where = where + " and password like ?";
                sqlParamValueList.add("%" + dto.getPassword() + "%");
                pageDTO.getQueryParam().put("password",
dto.getPassword());
            }
            if (dto.getBeginAge() != -1) {
                where = where + " and age >= ?";
                sqlParamValueList.add(dto.getBeginAge());
                pageDTO.getQueryParam().put("beginAge",
dto.getBeginAge());
            }
            if (dto.getEndAge() != -1) {
                where = where + " and age <= ?";
                sqlParamValueList.add(dto.getEndAge());
                pageDTO.getQueryParam().put("endAge",
dto.getEndAge());
            }
            if (dto.getBeginInsertDate() != null) {
                where = where + " and insertdate >= ?";
                sqlParamValueList.add(dto.getBeginInsertDate());
                pageDTO.getQueryParam().put("beginInsertDate",
dto.getBeginInsertDate());
            }
            if (dto.getEndInsertDate() != null) {
                where = where + " and insertdate <= ?";
                sqlParamValueList.add(dto.getEndInsertDate());
                pageDTO.getQueryParam().put("endInsertDate",
dto.getEndInsertDate());
            }

            sql1 = sql1 + where;
            sql2 = sql2 + where + order + limit;

            System.out.println(sql1);
            System.out.println(sql2);
```

```
            List queryPageList = new ArrayList();
            {
                Connection conn = GetConnection.getConnection();
                PreparedStatement ps = conn.prepareStatement(sql1);
                //将查询的参数放入 List 中，
                //然后通过 for 循环依次对 List 中的值与对应的?进行对应传参。
                for (int i = 0; i < sqlParamValueList.size(); i++) {
                    ps.setObject(i + 1, sqlParamValueList.get(i));
                }
                ResultSet rs = ps.executeQuery();
                while (rs.next()) {
                    int recordCount = rs.getInt("recordCount");
                    pageDTO.getGotoPage().put("recordCount",
recordCount);
                }
                rs.close();
                ps.close();
                conn.close();
            }
            {
                Connection conn = GetConnection.getConnection();
                PreparedStatement ps = conn.prepareStatement(sql2);
                for (int i = 0; i < sqlParamValueList.size(); i++) {
                    ps.setObject(i + 1, sqlParamValueList.get(i));
                }
                ResultSet rs = ps.executeQuery();
                while (rs.next()) {
                    int id = rs.getInt("id");
                    String username = rs.getString("username");
                    String password = rs.getString("password");
                    int age = rs.getInt("age");
                    Date insertdate = rs.getDate("insertdate");

                    Userinfo userinfo = new Userinfo();
                    userinfo.setId(id);
                    userinfo.setUsername(username);
                    userinfo.setPassword(password);
                    userinfo.setAge(age);
                    userinfo.setInsertdate(insertdate);
                    listUserinfo.add(userinfo);

                    queryPageList.add(userinfo);
                }
                rs.close();
```

```
                ps.close();
                conn.close();
            }
            pageDTO.setPageList(queryPageList);
            System.out.println(queryPageList.size());
        } catch (
                SQLException e) {
            e.printStackTrace();
        }
        return pageDTO;
    }
}
```

1.20.4.3 Service 类代码

```
package service;

import dao.UserinfoDAO;
import dto.PageDTO;
import dto.UserinfoPageQueryDTO;

public class UserinfoService {
    private UserinfoDAO userinfoDao = new UserinfoDAO();

    public PageDTO selectUserinfoPage(UserinfoPageQueryDTO dto) {
        PageDTO pageDTO = userinfoDao.selectUserinfoPage(dto);
        return pageDTO;
    }
}
```

1.20.4.4 实体类代码

```
package entity;

import java.util.Date;

public class Userinfo {
    private int id;
    private String username;
    private String password;
    private int age;
    private Date insertdate;

    public Userinfo() {
```

```
        this.id = id;
        this.username = username;
        this.password = password;
        this.age = age;
        this.insertdate = insertdate;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public Date getInsertdate() {
        return insertdate;
    }

    public void setInsertdate(Date insertdate) {
```

```
        this.insertdate = insertdate;
    }
}
```

1.20.4.5 2 个 DTO 类代码

```
package dto;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class PageDTO {
    //保存查询条件的值，目的是在前台重新显示
    private Map queryParam = new HashMap();
    //查询的分页数据列表
    private List pageList;
    //分页导航，首页，上一页，下一页，末页，总页数等相关的数值
    private Map gotoPage = new HashMap();

    public Map getQueryParam() {
        return queryParam;
    }

    public void setQueryParam(Map queryParam) {
        this.queryParam = queryParam;
    }

    public List getPageList() {
        return pageList;
    }

    public void setPageList(List pageList) {
        this.pageList = pageList;
    }

    public Map getGotoPage() {
        return gotoPage;
    }

    public void setGotoPage(Map gotoPage) {
        this.gotoPage = gotoPage;
```

```
    }
}

package dto;

import java.util.Date;

public class UserinfoPageQueryDTO {

    private String username;
    private String password;
    private int beginAge;
    private int endAge;
    private Date beginInsertDate;
    private Date endInsertDate;
    private int gotoPage;

    public UserinfoPageQueryDTO() {
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public int getBeginAge() {
        return beginAge;
    }

    public void setBeginAge(int beginAge) {
        this.beginAge = beginAge;
    }
```

```
    public int getEndAge() {
        return endAge;
    }

    public void setEndAge(int endAge) {
        this.endAge = endAge;
    }

    public Date getBeginInsertDate() {
        return beginInsertDate;
    }

    public void setBeginInsertDate(Date beginInsertDate) {
        this.beginInsertDate = beginInsertDate;
    }

    public Date getEndInsertDate() {
        return endInsertDate;
    }

    public void setEndInsertDate(Date endInsertDate) {
        this.endInsertDate = endInsertDate;
    }

    public int getGotoPage() {
        return gotoPage;
    }

    public void setGotoPage(int gotoPage) {
        this.gotoPage = gotoPage;
    }
}
```

1.20.4 laypage 结合 table 实现服务端分页-有刷新

Servlet类代码如下：

```
package controller;

import dto.PageDTO;
import dto.UserinfoPageQueryDTO;
import service.UserinfoService;

import javax.servlet.ServletException;
```

```
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

@WebServlet(name = "ListUserinfo", urlPatterns = "/ListUserinfo")
public class ListUserinfo extends HttpServlet {
    protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException
{
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        String beginAge = request.getParameter("beginAge");
        String endAge = request.getParameter("endAge");
        String beginInsertDate =
request.getParameter("beginInsertDate");
        String endInsertDate = request.getParameter("endInsertDate");
        String gotoPage = request.getParameter("gotoPage");

        System.out.println(username);
        System.out.println(password);
        System.out.println(beginAge);
        System.out.println(endAge);
        System.out.println(beginInsertDate);
        System.out.println(endInsertDate);

        int beginAgeInt = -1;
        int endAgeInt = -1;
        int gotoPageInt = 1;
        if (!"".equals(beginAge) && beginAge != null) {
            try {
                beginAgeInt = Integer.parseInt(beginAge);
            } catch (NumberFormatException e) {
                beginAgeInt = -1;
            }
        }
        if (!"".equals(endAge) && endAge != null) {
            try {
                endAgeInt = Integer.parseInt(endAge);
            } catch (NumberFormatException e) {
```

```
            endAgeInt = -1;
        }
    }
    if (!"".equals(gotoPage) && gotoPage != null) {
        try {
            gotoPageInt = Integer.parseInt(gotoPage);
        } catch (NumberFormatException e) {
            gotoPageInt = 1;
        }
    }

    Date beginInsertDateObject = null;
    Date endInsertDateObject = null;
    SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");
    if (!"".equals(beginInsertDate) && beginInsertDate != null) {
        try {
            beginInsertDateObject = format.parse(beginInsertDate);
        } catch (ParseException e) {
            beginInsertDateObject = null;
        }
    }
    if (!"".equals(endInsertDate) && endInsertDate != null) {
        try {
            endInsertDateObject = format.parse(endInsertDate);
        } catch (ParseException e) {
            endInsertDateObject = null;
        }
    }

    UserinfoPageQueryDTO dto = new UserinfoPageQueryDTO();
    dto.setUsername(username);
    dto.setPassword(password);
    dto.setBeginAge(beginAgeInt);
    dto.setEndAge(endAgeInt);
    dto.setBeginInsertDate(beginInsertDateObject);
    dto.setEndInsertDate(endInsertDateObject);
    dto.setGotoPage(gotoPageInt);

    UserinfoService userinfoService = new UserinfoService();
    PageDTO pageDTO = userinfoService.selectUserinfoPage(dto);
    request.setAttribute("pageDTO", pageDTO);

request.getRequestDispatcher("listUserinfo.jsp").forward(request,
response);
```

```
    }
}
```

index.jsp文件代码如下：

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
    <head>
        <title>Title</title>
    </head>
    <body>
        <jsp:forward page="/ListUserinfo"></jsp:forward>
    </body>
</html>
```

listUserinfo.jsp文件代码如下：

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<html>
    <head>
        <title>Title</title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-row layui-col-space10">
            <div class="layui-col-md12">
                <div>
                    <input type="hidden" id="gotoPage"
name="gotoPage"/>
                    <table class="layui-table">
                        <colgroup>
                            <col width="100">
                            <col>
                            <col width="100">
                            <col>
                            <col width="100">
                            <col>
                            <col width="100">
                            <col>
                        </colgroup>
                        <tbody>
                            <tr>
```

```
                            <td>账号</td>
                            <td><input type="text" id="username"
name="username" placeholder="请输入账号"
                                       autocomplete="off"
                                       class="layui-input"
value="${pageDTO.queryParam.username}"></td>
                            <td>密码</td>
                            <td><input type="text" id="password"
name="password" placeholder="请输入账号"
                                       autocomplete="off"
                                       class="layui-input"
value="${pageDTO.queryParam.password}"></td>
                            <td>开始年龄</td>
                            <td><input type="text" id="beginAge"
name="beginAge" placeholder="请输入账号"
                                       autocomplete="off"
                                       class="layui-input"
value="${pageDTO.queryParam.beginAge}"></td>
                            <td>结束年龄</td>
                            <td><input type="text" id="endAge"
name="endAge" placeholder="请输入账号"
                                       autocomplete="off"
                                       class="layui-input"
value="${pageDTO.queryParam.endAge}"></td>
                        </tr>
                        <tr>
                            <td>开始日期</td>
                            <td><input type="text"
id="beginInsertDate" name="beginInsertDate"
                                       placeholder="请输入账号"
autocomplete="off"
                                       class="layui-input"
                                       value="<fmt:formatDate
value="${pageDTO.queryParam.beginInsertDate}"
pattern="yyyy-MM-dd"></fmt:formatDate>">
                            </td>
                            <td>结束日期</td>
                            <td><input type="text"
id="endInsertDate" name="endInsertDate" placeholder="请输入账号"
                                       autocomplete="off"
                                       class="layui-input"
                                       value="<fmt:formatDate
value="${pageDTO.queryParam.endInsertDate}"
pattern="yyyy-MM-dd"></fmt:formatDate>">
```

```
                            </td>
                            <td></td>
                            <td></td>
                            <td></td>
                            <td></td>
                        </tr>
                        <tr>
                            <td colspan="8" style="text-align:
center;">
                                <button id="queryButton"
type="button" class="layui-btn layui-btn-normal"
                                        style="width: 100px;">
                                    查询
                                </button>
                                <button id="resetButton" type="reset"
class="layui-btn layui-btn-warm"
                                        style="width: 100px;">重置
                                </button>
                            </td>
                        </tr>
                    </tbody>
                </table>
            </div>
        </div>
    </div>
    <div class="layui-row layui-col-space10" style="height:
700px;">
        <div class="layui-col-md12">
            <div>
                <table class="layui-table" lay-even lay-size="lg">
                    <colgroup>
                        <col width="100">
                        <col width="100">
                        <col>
                        <col>
                        <col width="100">
                        <col width="150">
                    </colgroup>
                    <thead>
                        <tr>
                            <th>序号</th>
                            <th>ID</th>
                            <th>账号</th>
                            <th>密码</th>
```

```
                            <th>年龄</th>
                            <th>注册日期</th>
                        </tr>
                    </thead>
                    <tbody>
                        <c:forEach var="eachUserinfo"
items="${pageDTO.pageList}" varStatus="status">
                            <tr>
                                <td>
                                    <c:if
test="${param.gotoPage==null}">
                                        ${status.index+1}
                                    </c:if>
                                    <c:if
test="${param.gotoPage!=null}">

${(param.gotoPage-1)*12+status.index+1}
                                    </c:if>
                                </td>
                                <td>${eachUserinfo.id}</td>
                                <td>${eachUserinfo.username}</td>
                                <td>${eachUserinfo.password}</td>
                                <td>${eachUserinfo.age}</td>
                                <td>${eachUserinfo.insertdate}</td>
                            </tr>
                        </c:forEach>
                    </tbody>
                </table>
            </div>
        </div>
    </div>
    <div class="layui-row layui-col-space10">
        <div class="layui-col-md12">
            <div>
                <table class="layui-table">
                    <thead>
                        <tr>
                            <th style="background-color:
white;text-align: center">
                                <div id="page"></div>
                            </th>
                        </tr>
                    </thead>
                    </tbody>
```

```
                </table>
            </div>
        </div>
    </div>
    <div id="test1"></div>
    <script>
        var laypage = layui.laypage;
        var element = layui.element;
        var form = layui.form;
        var laydate = layui.laydate;

        laydate.render({
            elem: '#beginInsertDate' //指定元素
        });
        laydate.render({
            elem: '#endInsertDate' //指定元素
        });

        function beginQuery(gotoPage) {
            var paramValue = "ListUserinfo?gotoPage=" +
$("#gotoPage").val();
            paramValue = paramValue + "&" + "username=" +
$("#username").val();
            paramValue = paramValue + "&" + "password=" +
$("#password").val();
            paramValue = paramValue + "&" + "beginAge=" +
$("#beginAge").val();
            paramValue = paramValue + "&" + "endAge=" +
$("#endAge").val();
            paramValue = paramValue + "&" + "beginInsertDate=" +
$("#beginInsertDate").val();
            paramValue = paramValue + "&" + "endInsertDate=" +
$("#endInsertDate").val();
            paramValue = paramValue + "#!laypageValue=" + gotoPage;
            location.href = paramValue;
        }

        laypage.render({
            elem: 'page',
            count: ${pageDTO.gotoPage.recordCount},
            limit: 12,
            theme: "#1E9FFF",
            layout: ['prev', 'page', 'next', 'skip', 'count'],
            curr: location.hash.replace('#!laypageValue=', ''),
```

```
                hash: 'laypageValue',
                jump: function (obj, first) {
                    //obj 包含了当前分页的所有参数，比如：
                    //alert(obj.curr); //得到当前页，以便向服务端请求对应页
的数据。
                    //alert(obj.limit); //得到每页显示的条数
                    //首次不执行
                    if (!first) {
                        $("#gotoPage").val(obj.curr);
                        beginQuery($("#gotoPage").val());
                    } else {
                        //alert("first run begin query!");
                    }
                }
            });
            $(document).ready(function () {
                $("#queryButton").click(function () {
                    $("#gotoPage").val(1);
                    beginQuery($("#gotoPage").val());
                });

                $("#resetButton").click(function () {
                    $("#gotoPage").val(1);
                    $("#username").val("")
                    $("#password").val("");
                    $("#beginAge").val("");
                    $("#endAge").val("");
                    $("#beginInsertDate").val("");
                    $("#endInsertDate").val("");
                });
            });

            form.render();
        </script>
    </body>
</html>
```

1.20.5 laypage 结合 table 实现服务端分页-无刷新

Servlet类代码如下：

```
package controller;

import com.fasterxml.jackson.databind.ObjectMapper;
```

```
import dto.PageDTO;
import dto.UserinfoPageQueryDTO;
import service.UserinfoService;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

@WebServlet(name = "ListUserinfo", urlPatterns = "/ListUserinfo")
public class ListUserinfo extends HttpServlet {
    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException
{
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        String beginAge = request.getParameter("beginAge");
        String endAge = request.getParameter("endAge");
        String beginInsertDate =
request.getParameter("beginInsertDate");
        String endInsertDate = request.getParameter("endInsertDate");
        String gotoPage = request.getParameter("gotoPage");

        System.out.println(username);
        System.out.println(password);
        System.out.println(beginAge);
        System.out.println(endAge);
        System.out.println(beginInsertDate);
        System.out.println(endInsertDate);

        int beginAgeInt = -1;
        int endAgeInt = -1;
        int gotoPageInt = 1;
        if (!"".equals(beginAge) && beginAge != null) {
            try {
                beginAgeInt = Integer.parseInt(beginAge);
            } catch (NumberFormatException e) {
                beginAgeInt = -1;
```

```
    }
}
if (!"".equals(endAge) && endAge != null) {
    try {
        endAgeInt = Integer.parseInt(endAge);
    } catch (NumberFormatException e) {
        endAgeInt = -1;
    }
}
if (!"".equals(gotoPage) && gotoPage != null) {
    try {
        gotoPageInt = Integer.parseInt(gotoPage);
    } catch (NumberFormatException e) {
        gotoPageInt = 1;
    }
}

Date beginInsertDateObject = null;
Date endInsertDateObject = null;
SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");
if (!"".equals(beginInsertDate) && beginInsertDate != null) {
    try {
        beginInsertDateObject = format.parse(beginInsertDate);
    } catch (ParseException e) {
        beginInsertDateObject = null;
    }
}
if (!"".equals(endInsertDate) && endInsertDate != null) {
    try {
        endInsertDateObject = format.parse(endInsertDate);
    } catch (ParseException e) {
        endInsertDateObject = null;
    }
}

UserinfoPageQueryDTO dto = new UserinfoPageQueryDTO();
dto.setUsername(username);
dto.setPassword(password);
dto.setBeginAge(beginAgeInt);
dto.setEndAge(endAgeInt);
dto.setBeginInsertDate(beginInsertDateObject);
dto.setEndInsertDate(endInsertDateObject);
dto.setGotoPage(gotoPageInt);
```

```
        UserinfoService userinfoService = new UserinfoService();
        PageDTO pageDTO = userinfoService.selectUserinfoPage(dto);

        ObjectMapper mapper = new ObjectMapper();
        String jsonString = mapper.writeValueAsString(pageDTO);
        System.out.println(jsonString);

        response.setContentType("text/html;charset=utf-8");
        PrintWriter out = response.getWriter();
        out.print(jsonString);
        out.flush();
        out.close();
    }
}
```

index.jsp文件代码如下：

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
    <head>
        <title>Title</title>
    </head>
    <body>
        <jsp:forward page="/listUserinfo.html"></jsp:forward>
    </body>
</html>
```

listUserinfo.html文件代码如下：

```
<html>
    <head>
        <title>Title</title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-row layui-col-space10">
            <div class="layui-col-md12">
                <div>
                    <input type="hidden" id="gotoPage"
name="gotoPage"/>
                    <table class="layui-table">
                        <colgroup>
                            <col width="100">
                            <col>
```

```
                        <col width="100">
                        <col>
                        <col width="100">
                        <col>
                        <col width="100">
                        <col>
                    </colgroup>
                    <tbody>
                        <tr>
                            <td>账号</td>
                            <td><input type="text" id="username"
name="username" placeholder="请输入账号"
                                       autocomplete="off"
                                       class="layui-input"></td>
                            <td>密码</td>
                            <td><input type="text" id="password"
name="password" placeholder="请输入账号"
                                       autocomplete="off"
                                       class="layui-input"></td>
                            <td>开始年龄</td>
                            <td><input type="text" id="beginAge"
name="beginAge" placeholder="请输入账号"
                                       autocomplete="off"
                                       class="layui-input"></td>
                            <td>结束年龄</td>
                            <td><input type="text" id="endAge"
name="endAge" placeholder="请输入账号"
                                       autocomplete="off"
                                       class="layui-input"></td>
                        </tr>
                        <tr>
                            <td>开始日期</td>
                            <td><input type="text"
id="beginInsertDate" name="beginInsertDate"
                                       placeholder="请输入账号"
autocomplete="off"
                                       class="layui-input"/>
                            </td>
                            <td>结束日期</td>
                            <td><input type="text"
id="endInsertDate" name="endInsertDate" placeholder="请输入账号"
                                       autocomplete="off"
                                       class="layui-input"/>
                            </td>
```

```
                            <td></td>
                            <td></td>
                            <td></td>
                            <td></td>
                        </tr>
                        <tr>
                            <td colspan="8" style="text-align:
center;">
                                <button id="queryButton"
type="button" class="layui-btn layui-btn-normal"
                                        style="width: 100px;">
                                    查询
                                </button>
                                <button id="resetButton" type="reset"
class="layui-btn layui-btn-warm"
                                        style="width: 100px;">重置
                                </button>
                            </td>
                        </tr>
                    </tbody>
                </table>
            </div>
        </div>
    </div>
    <div class="layui-row layui-col-space10" style="height:
700px;">
        <div class="layui-col-md12">
            <div>
                <table id="userinfoPageTable" class="layui-table"
lay-even lay-size="lg">
                    <colgroup>
                        <col width="100">
                        <col width="100">
                        <col>
                        <col>
                        <col width="100">
                        <col width="150">
                    </colgroup>
                    <thead>
                        <tr>
                            <th>序号</th>
                            <th>ID</th>
                            <th>账号</th>
                            <th>密码</th>
```

```
                        <th>年龄</th>
                        <th>注册日期</th>
                    </tr>
                </thead>
                <tbody>
                </tbody>
            </table>
        </div>
    </div>
</div>
<div class="layui-row layui-col-space10">
    <div class="layui-col-md12">
        <div>
            <table class="layui-table">
                <thead>
                    <tr>
                        <th style="background-color:
white;text-align: center">
                            <div id="page"></div>
                        </th>
                    </tr>
                </thead>
                <tbody>
            </table>
        </div>
    </div>
</div>
<div id="test1"></div>
<script>
    var laypage = layui.laypage;
    var element = layui.element;
    var form = layui.form;
    var laydate = layui.laydate;

    laydate.render({
        elem: '#beginInsertDate' //指定元素
    });
    laydate.render({
        elem: '#endInsertDate' //指定元素
    });

    function createTR(eachUserinfo, index) {
        var newTR = $("<tr></tr>");
```

```
            var td_number = $("<td></td>");
            var gotoPage = $("#gotoPage").val();
            $(td_number).text((gotoPage - 1) * 12 + index + 1);
            var td_id = $("<td></td>");
            $(td_id).text(eachUserinfo.id);
            var td_username = $("<td></td>");
            $(td_username).text(eachUserinfo.username);
            var td_password = $("<td></td>");
            $(td_password).text(eachUserinfo.password);
            var td_age = $("<td></td>");
            $(td_age).text(eachUserinfo.age);
            var td_insertdate = $("<td></td>");
            var insertDate = new Date(eachUserinfo.insertdate);
            var showDateString = insertDate.getFullYear() + "-" +
(insertDate.getMonth() + 1) + "-" + insertDate.getDate();
            $(td_insertdate).text(showDateString);

            $(newTR).append(td_number);
            $(newTR).append(td_id);
            $(newTR).append(td_username);
            $(newTR).append(td_password);
            $(newTR).append(td_age);
            $(newTR).append(td_insertdate);

            return newTR;
        }

        function fillPageInfo(gotoPage) {
            $("#userinfoPageTable").find("tbody").empty();
            $.post("ListUserinfo?t=" + new Date().getTime(), {
                "username": $("#username").val(),
                "password": $("#password").val(),
                "beginAge": $("#beginAge").val(),
                "endAge": $("#endAge").val(),
                "beginInsertDate": $("#beginInsertDate").val(),
                "endInsertDate": $("#endInsertDate").val(),
                "gotoPage": gotoPage
            }, function (data) {
                var jsonObject = JSON.parse(data);
                var pageList = jsonObject.pageList;
                for (var i = 0; i < pageList.length; i++) {
                    var returnNewTR = createTR(pageList[i], i);

$("#userinfoPageTable").find("tbody").append(returnNewTR);
```

```
                }

                laypage.render({
                    elem: 'page',
                    count: jsonObject.gotoPage.recordCount,
                    limit: 12,
                    theme: "#1e9fff",
                    layout: ['prev', 'page', 'next', 'skip',
'count'],
                    curr: location.hash.replace('#!laypageValue=',
''),
                    hash: 'laypageValue',
                    jump: function (obj, first) {
                        //obj 包含了当前分页的所有参数，比如：
                        //alert(obj.curr); //得到当前页，以便向服务端请
求对应页的数据。
                        //alert(obj.limit); //得到每页显示的条数
                        //首次不执行
                        if (!first) {
                            $("#gotoPage").val(obj.curr);
                            fillPageInfo($("#gotoPage").val());
                        } else {
                            //alert("first run begin query!");
                        }
                    }
                });
            });
        }

        $(document).ready(function () {
            $("#queryButton").click(function () {
                $("#gotoPage").val(1);
                fillPageInfo($("#gotoPage").val());
            });

            $("#resetButton").click(function () {
                $("#gotoPage").val(1);
                $("#username").val("")
                $("#password").val("");
                $("#beginAge").val("");
                $("#endAge").val("");
                $("#beginInsertDate").val("");
                $("#endInsertDate").val("");
            });
```

```
            $("#gotoPage").val(1);
            fillPageInfo(1);
        });

        form.render();
    </script>
</body>
</html>
```

1.21 常用 element 元素操作

element 组件包含导航，选项卡，进度条，面板。这些 element 功能的开启只需要加载 element 模块即会自动完成，不用跟其它模块一样为某一个功能而去调用一个方法。本章节介绍使用 Layui 提供的 API 操作 element 组件。

模块加载名称：element。

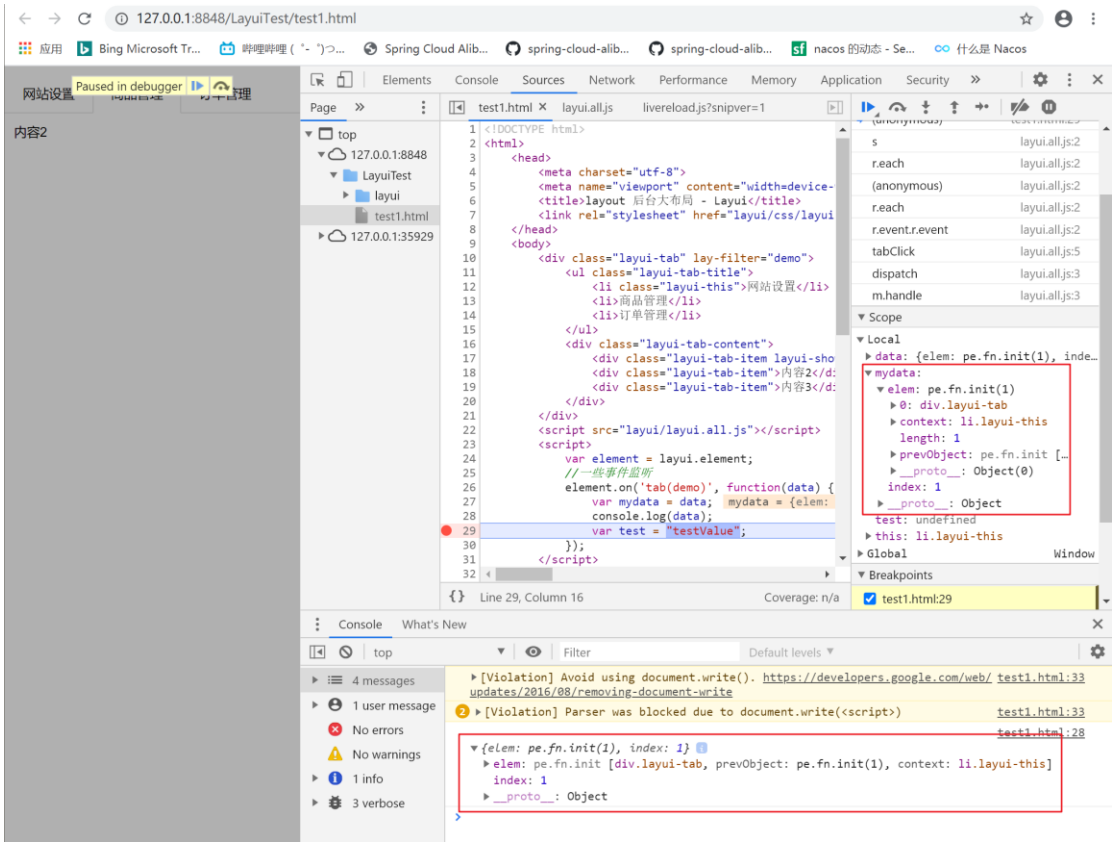
1.21.1 基本使用

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
    </head>
    <body>
        <div class="layui-tab" lay-filter="demo">
            <ul class="layui-tab-title">
                <li class="layui-this">网站设置</li>
                <li>商品管理</li>
                <li>订单管理</li>
            </ul>
            <div class="layui-tab-content">
                <div class="layui-tab-item layui-show">内容1</div>
                <div class="layui-tab-item">内容2</div>
                <div class="layui-tab-item">内容3</div>
            </div>
        </div>
        <script src="layui/layui.all.js"></script>
```

```
        <script>
            var element = layui.element;
            //一些事件监听
            element.on('tab(demo)', function(data) {
                var mydata = data;
                console.log(data);
                var test = "testValue";
            });
        </script>

    </body>
</html>
```

运行结果如图 1-xx 所示。



1.21.2 预设元素属性

通过自定义元素属性来作为元素的功能参数，它们一般配置在容器外层，例如：

…

element 模块支持的元素属性如图 1-xx 所示。

属性名	可选值	说明
lay-filter	任意字符	事件过滤器（公用属性），主要用于事件的精确匹配，跟选择器是比较类似的。
lay-allowClose	true	针对于Tab容器，是否允许选项卡关闭。默认不允许，即不用设置该属性
lay-separator	任意分隔符	针对于面包屑容器

1.21.3 基础方法

基础方法允许在外部主动对元素进行操作。

1.21.3.1 方法 var element = layui.element;

返回的 element 变量为该实例的对象，携带一些用于元素操作的基础方法。

测试代码如下：

```
<script>
    var element = layui.element;
</script>
```

1.21.3.2 方法 element.on(filter, callback);

用于元素的事件监听。

语法：element.on('event(过滤器值)', callback);

element 模块在 layui 事件机制中注册了 element 模块事件，目前 element 模块所支持的事件名称如图 1-xx 所示。

event	描述
tab	监听 Tab 选项卡切换事件
tabDelete	监听 Tab 监听选项卡删除事件
nav	监听导航菜单的点击事件
collapse	监听折叠面板展开或收缩事件

所以当使用 layui.onevent()自定义事件时，请勿与上面系统提供的事件名称相同。

默认情况下，事件所监听的是全部的元素，但如果只想监听某一个元素，使用事件过滤器即可，例如：

结合 javascript 代码：

```
element.on('tab(test)', function(data){
   console.log(data);
```

```
});
```

1.21.3.2.1 监听选项卡切换 tab

tab 选项卡点击切换时触发，回调函数返回一个 object 参数，携带两个成员。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-tab" lay-filter="mytab">
            <ul class="layui-tab-title">
                <li class="layui-this">网站设置</li>
                <li>用户管理</li>
                <li>权限分配</li>
                <li>商品管理</li>
                <li>订单管理</li>
            </ul>
            <div class="layui-tab-content">
                <div class="layui-tab-item layui-show">内容1</div>
                <div class="layui-tab-item">内容2</div>
                <div class="layui-tab-item">内容3</div>
                <div class="layui-tab-item">内容4</div>
                <div class="layui-tab-item">内容5</div>
            </div>
        </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;

            element.on('tab(mytab)', function(data) {
                console.log(this); //当前Tab标题所在的原始DOM元素
                console.log(data.index); //得到当前Tab的所在下标
                console.log(data.elem); //得到当前的Tab大容器
            });
        </script>
```

```
    </body>
</html>
```

1.21.3.2.2 监听选项卡删除 tabDelete

tab 选项卡被删除时触发，回调函数返回一个 object 参数，携带两个成员。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-tab" lay-allowClose="true" lay-filter="mytab">
            <ul class="layui-tab-title">
                <li class="layui-this">网站设置</li>
                <li>用户基本管理</li>
                <li>权限分配</li>
                <li>全部历史商品管理文字长一点试试</li>
                <li>订单管理</li>
            </ul>
            <div class="layui-tab-content">
                <div class="layui-tab-item layui-show">1</div>
                <div class="layui-tab-item">2</div>
                <div class="layui-tab-item">3</div>
                <div class="layui-tab-item">4</div>
                <div class="layui-tab-item">5</div>
                <div class="layui-tab-item">6</div>
            </div>
        </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            element.on('tabDelete(mytab)', function(data) {
                console.log(this); //当前Tab标题所在的原始DOM元素
                console.log(data.index); //得到当前Tab的所在下标
                console.log(data.elem); //得到当前的Tab大容器
```

```
            });
        </script>
    </body>
</html>
```

1.21.3.2.3 监听导航菜单的点击 nav

当点击导航父级菜单和二级菜单时触发，回调函数返回所点击菜单的 DOM 对象。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <ul class="layui-nav" lay-filter="mynav">
            <li class="layui-nav-item"><a href="#">最新活动</a></li>
            <li class="layui-nav-item layui-this"><a href="#">产品</a></li>
            <li class="layui-nav-item"><a href="#">大数据</a></li>
            <li class="layui-nav-item">
                <a href="#">解决方案</a>
                <dl class="layui-nav-child">
                    <!-- 二级菜单 -->
                    <dd><a href="#">移动模块</a></dd>
                    <dd><a href="#">后台模版</a></dd>
                    <dd><a href="#">电商平台</a></dd>
                </dl>
            </li>
            <li class="layui-nav-item"><a href="#">社区</a></li>
        </ul>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            element.on('nav(mynav)', function(elem) {
                console.log(elem); //得到当前点击的DOM对象
            });
        </script>
```

```
    </body>
</html>
```

1.21.3.2.4 监听折叠面板 collapse

当对折叠面板点击展开或收缩时触发，回调函数返回一个 object 参数，携带三个成员。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-collapse" lay-filter="mycollapse">
            <div class="layui-colla-item">
                <h2 class="layui-colla-title">杜甫</h2>
                <div class="layui-colla-content layui-show">内容区域1</div>
            </div>
            <div class="layui-colla-item">
                <h2 class="layui-colla-title">李清照</h2>
                <div class="layui-colla-content layui-show">内容区域2</div>
            </div>
            <div class="layui-colla-item">
                <h2 class="layui-colla-title">鲁迅</h2>
                <div class="layui-colla-content">内容区域3</div>
            </div>
        </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            element.on('collapse(mycollapse)', function(data) {
                console.log(data.show); //得到当前面板的展开状态，true或者
false
                console.log(data.title); //得到当前点击面板的标题区域DOM对象
                console.log(data.content); //得到当前点击面板的内容区域DOM对
象
            });
```

```
        </script>
    </body>
</html>
```

1.21.3.3 方法 element.tabAdd(filter, options);

用于新增一个 tab 选项标签。
参数 filter：tab 元素的 lay-filter="value"过滤器的值（value）。
参数 options：设置 options 参数对象，目前支持的 options 选项如下示例：

```
element.tabAdd('demo', {
  title: '选项卡的标题',
  content: '选项卡的内容', //支持传入 html
  id: '选项卡标题的 lay-id 属性值'
});
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-tab" lay-filter="mytab">
            <ul class="layui-tab-title">
                <li class="layui-this">网站设置</li>
                <li>用户管理</li>
                <li>权限分配</li>
                <li>商品管理</li>
                <li>订单管理</li>
            </ul>
            <div class="layui-tab-content">
                <div class="layui-tab-item layui-show">内容1</div>
                <div class="layui-tab-item">内容2</div>
                <div class="layui-tab-item">内容3</div>
                <div class="layui-tab-item">内容4</div>
                <div class="layui-tab-item">内容5</div>
            </div>
        </div>
```

```
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            $(document).ready(function() {
                $("#addTabPage").click(function() {
                    element.tabAdd('mytab', {
                        title: '选项卡的标题',
                        content: '选项卡的内容', //支持传入html
                        id: '8888'
                    });
                });
            });
    </script>
        <input type="button" id="addTabPage" value="addTabPage" />
    </body>
</html>
```

1.21.3.4 方法 element.tabDelete(filter, layid);

用于删除指定的 tab 选项。
参数 filter：tab 元素的 lay-filter="value"过滤器的值（value）。
参数 layid：选项卡标题列表的 lay-id 属性值。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-tab" lay-filter="mytab">
            <ul class="layui-tab-title">
                <li class="layui-this" lay-id="111">文章列表</li>
                <li lay-id="222">发送信息</li>
                <li lay-id="333">权限分配</li>
                <li lay-id="444">审核</li>
                <li lay-id="555">订单管理</li>
            </ul>
```

```
            <div class="layui-tab-content">
                <div class="layui-tab-item layui-show">1</div>
                <div class="layui-tab-item">2</div>
                <div class="layui-tab-item">3</div>
                <div class="layui-tab-item">4</div>
                <div class="layui-tab-item">5</div>
            </div>
        </div>

        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            $(document).ready(function() {
                $("#removeTabPage").click(function() {
                    element.tabDelete('mytab', '444');
                    //删除 lay-id="444" 的这一项
                });
            });
    </script>
        <input type="button" id="removeTabPage" value="removeTabPage" />
    </body>
</html>
```

1.21.3.5 方法 element.tabChange(filter, layid);

用于外部切换到指定的 tab 标签页上，示例代码如下：
element.tabChange('demo', 'layid'); //切换到 lay-id="yyy"的这一项

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-tab" lay-filter="mytab">
            <ul class="layui-tab-title">
                <li class="layui-this" lay-id="111">文章列表</li>
```

```
                <li lay-id="222">发送信息</li>
                <li lay-id="333">权限分配</li>
                <li lay-id="444">审核</li>
                <li lay-id="555">订单管理</li>
            </ul>
            <div class="layui-tab-content">
                <div class="layui-tab-item layui-show">1</div>
                <div class="layui-tab-item">2</div>
                <div class="layui-tab-item">3</div>
                <div class="layui-tab-item">4</div>
                <div class="layui-tab-item">5</div>
            </div>
        </div>

        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            $(document).ready(function() {
                $("#addTabPage").click(function() {
                    element.tabAdd('mytab', {
                        title: '选项卡的标题',
                        content: '选项卡的内容', //支持传入html
                        id: '8888'
                    });
                    element.tabChange('mytab', '8888');
                });
            });
    </script>
        <input type="button" id="addTabPage" value="addTabPage" />
    </body>
</html>
```

1.21.3.6 方法 element.progress(filter, percent);

用于动态改变进度条百分比，示例代码如下：

element.progress('demo', '30%');

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1,
```

```
maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <div class="layui-progress" lay-filter="myprogress">
            <div class="layui-progress-bar" lay-percent="50%"></div>
        </div>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;
            $(document).ready(function() {
                $("#setprogress").click(function() {
                    element.progress('myprogress',"99%");
                });
            });
        </script>
        <input type="button" id="setprogress" value="setprogress" />
    </body>
</html>
```

1.21.4 更新渲染

跟表单元素一样，很多时候页面元素可能是动态生成的，这时 element 的相关功能将不会对其有效，所以必须手工执行 element.render(type, filter);方法。

第一个参数 type 是元素的 type 类型，可选。默认对全部类型的元素进行一次更新，可局部刷新的 type 值如图 1-xx 所示。

参数（type）值	描述
tab	重新对tab选项卡进行初始化渲染
nav	重新对导航进行渲染
breadcrumb	重新对面包屑进行渲染
progress	重新对进度条进行渲染
collapse	重新对折叠面板进行渲染

使用如下代码进行渲染：

element.render('nav');

第二个参数 filter 是元素的 lay-filter=""值，可以借助该参数完成对某一个元素进行渲染，而不是全部。示例代码如下：

【HTML】

```
<div class="layui-nav" lay-filter="test1">
  ...
</div>
<div class="layui-nav" lay-filter="test2">
  ...
</div>
```

【JavaScript】

```
//比如当对导航菜单 test1 动态插入了二级菜单，这时需要重新去对它进行渲染
element.render('nav', 'test1');
//对 lay-filter="test1"所在 nav 导航重新渲染。
//……
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
       <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
        <title>layout 后台大布局 - Layui</title>
        <link rel="stylesheet" href="layui/css/layui.css">
        <script src="js/jquery3.5.1.js"></script>
    </head>
    <body>
        <script src="layui/layui.all.js"></script>
        <script>
            var element = layui.element;

            function testMethod1(){
                var progressObject=$('<div class="layui-progress"><div class="layui-progress-bar" lay-percent="10%"></div></div>');
                $("body").append(progressObject);
                $("body").append("<br/><br/>");
            }

            function testMethod2(){
                var progressObject=$('<div class="layui-progress" lay-filter="myprogress"><div class="layui-progress-bar" lay-percent="90%"></div></div>');
                $("body").append(progressObject);
                element.render('progress', 'myprogress');
```

```
            }

            function testMethod3(){
                element.render('progress');
            }
        </script>
        <input type='button' onclick="javascript:testMethod1();"
value="button1">
        <br />
        <input type='button' onclick="javascript:testMethod2();"
value="button2">
        <br />
        <input type='button' onclick="javascript:testMethod3();"
value="button3">
        <br />
    </body>
</html>
```

如果 tab，nav，breadcrumb，progress，collapse 这 5 个组件是动态创建出来的，就需要进行更新 render 渲染。

1.22 模板引擎 laytpl

laytpl 是 JavaScript 模板引擎，在字符解析上有着比较出色的表现，欠缺之处在于异常调试上。

模块加载名称：laytpl。

在线调试：

http://www.layui.com/demo/laytpl.html

1.22.1 模板的快速使用

与一般的字符拼接不同的是，laytpl 模板可与数据分离，集中把逻辑处理放在 View 层，提升代码可维护性，尤其是针对大量模板渲染的情况。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
```

```
    <body>
        <script>
            var laytpl = layui.laytpl;

            //直接解析字符
            laytpl('{{ d.name }}是一位公猿').render({
                name: '贤心'
            }, function(string) {
                console.log(string); //贤心是一位公猿
            });

            //也可以采用下述同步写法，将render方法的回调函数剔除，可直接返回渲染好的字符
            var string = laytpl('{{ d.name }}是一位公猿').render({
                name: '贤心'
            });
            console.log(string); //贤心是一位公猿

            //如果模板较大，你也可以采用数据的写法，这样会比较直观一些
            var string = laytpl([
                '{{ d.name }}是一位公猿', '其它字符 {{ d.content }}  其它字符'
            ].join('')).render({
                name: '贤心',
                content: 'xyz',
            });
            console.log(string); //贤心是一位公猿
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.22.2 可以将模板存储在页面或其它任意位置

测试代码如下：

```
<!DOCTYPE html>
<html>
```

```
<head>
    <meta charset="utf-8">
    <title></title>
    <script src="js/jquery3.5.1.js"></script>
    <link rel="stylesheet" href="layui/css/layui.css" />
    <script src="layui/layui.all.js"></script>
</head>
<body>
    <!--第一步：编写模版。可以使用一个<script>标签存放模板，示例代码：-->
    <script id="demo" type="text/html">
      <h3>{{ d.title }}</h3>
      <ul>
      {{# layui.each(d.list, function(index, item){ }}
        <li>
          <span>{{ item.modname }}</span>
          <span>{{ item.alias }}</span>
          <span>{{ item.site || 'site为空' }}</span>
        </li>
      {{# }); }}
      {{# if(d.list.length === 0){ }}
        无数据
      {{# } }}
      </ul>
    </script>

    <!--第二步：建立视图。用于呈现渲染结果。-->
    <div id="view"></div>

    <!--第三步：渲染模版-->
    <script>
        var laytpl = layui.laytpl;

        var data = { //数据
            "title": "Layui常用模块",
            "list": [{
                "modname": "弹层",
                "alias": "layer",
                "site": "layer.layui.com"
            }, {
                "modname": "表单",
                "alias": "form"
            }]
        }
        var getTpl = demo.innerHTML;
```

```
            var view = document.getElementById('view');
            laytpl(getTpl).render(data, function(html) {
                view.innerHTML = html;
            });
        </script>
    </body>
</html>
```

以上代码的执行过程为：data+laytpl 模板=最终的 html 代码。

运行效果如图 1-xx 所示。

1.22.3 模版语法

模板语法如图 1-xx 所示。

语法	说明	示例
{{ d.field }}	输出一个普通字段，不转义html	code layui.code 01. {{ d.content }} 02.
{{= d.field }}	输出一个普通字段，并转义html	code layui.code 01. {{= d.title }} 02.
{{# JavaScript表达式 }}	JS 语句。一般用于逻辑处理。用分隔符加 # 号开头。 注意：如果你是想输出一个函数，正确的写法是：{{ fn() }}，而不是：{{# fn() }}	code layui.code 01. {{# 02. var fn = function(){ 03. return '2017-08-18'; 04. }; 05. }} 06. 07. {{# if(true){ }} 08. 开始日期：{{ fn() }} 09. {{# } else { }} 10. 已截止 11. {{# } }} 12.
{{! template !}}	对一段指定的模板区域进行过滤，即不解析该区域的模板。注：layui 2.1.6 新增	code layui.code 01. {{! 这里面的模板不会被解析 !}} 02.

1.22.3.1 转义{{ d.field }}

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <!--第一步：编写模版。可以使用一个<script>标签存放模板，示例代码：-->
        <script id="demo" type="text/html">
            <div>
                {{ d.htmlCode }}
            </div>
        </script>

        <!--第二步：建立视图。用于呈现渲染结果。-->
```

```
        <div id="view"></div>

        <!--第三步：渲染模版-->
        <script>
            var laytpl = layui.laytpl;

            var data = { //数据
                "htmlCode": "<input type='button' value='添加'/>"
            }
            var getTpl = demo.innerHTML;
            var view = document.getElementById('view');
            laytpl(getTpl).render(data, function(html) {
                view.innerHTML = html;
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

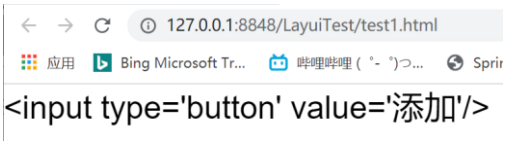
1.22.3.2 不转义{{= d.field }}

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <!--第一步：编写模版。可以使用一个<script>标签存放模板，示例代码：-->
        <script id="demo" type="text/html">
            <div>
                {{=d.htmlCode}}
```

```
            </div>
        </script>

        <!--第二步：建立视图。用于呈现渲染结果。-->
        <div id="view"></div>

        <!--第三步：渲染模版-->
        <script>
            var laytpl = layui.laytpl;

            var data = { //数据
                "htmlCode": "<input type='button' value='添加'/>"
            }
            var getTpl = demo.innerHTML;
            var view = document.getElementById('view');
            laytpl(getTpl).render(data, function(html) {
                view.innerHTML = html;
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



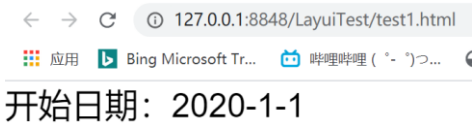
1.22.3.3 {{# JavaScript 表达式 }}

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <!--第一步：编写模版。可以使用一个<script>标签存放模板，示例代码：-->
        <script id="demo" type="text/html">
```

```
            {{#
              var fn = function(){
                return '2020-1-1';
              };
            }}

            {{#  if(true){ }}
              开始日期：{{ fn() }}
            {{#  } else { }}
              已截止
            {{#  } }}
        </script>

        <!--第二步：建立视图。用于呈现渲染结果。-->
        <div id="view"></div>

        <!--第三步：渲染模版-->
        <script>
            var laytpl = layui.laytpl;
            var getTpl = demo.innerHTML;
            var data = {};
            var view = document.getElementById('view');
            laytpl(getTpl).render(data, function(html) {
                view.innerHTML = html;
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



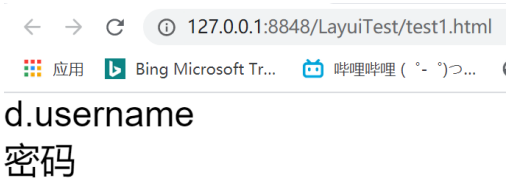
1.22.3.4 {{! template !}}忽略模板代码

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
```

```
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <!--第一步：编写模版。可以使用一个<script>标签存放模板，示例代码：-->
        <script id="demo" type="text/html">
            {{! d.username !}}
            <br/>
            {{ d.password }}
        </script>

        <!--第二步：建立视图。用于呈现渲染结果。-->
        <div id="view"></div>

        <!--第三步：渲染模版-->
        <script>
            var laytpl = layui.laytpl;
            var getTpl = demo.innerHTML;
            var data = {
                "usernane": "账号",
                "password": "密码"
            };
            var view = document.getElementById('view');
            laytpl(getTpl).render(data, function(html) {
                view.innerHTML = html;
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



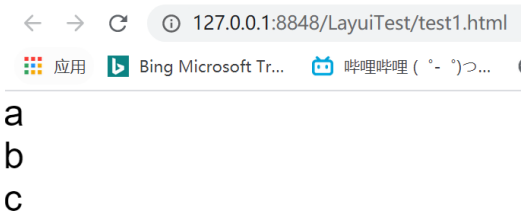
1.22.3.5 结合 for 循环

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
```

```
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <!--第一步：编写模版。可以使用一个<script>标签存放模板，示例代码：-->
        <script id="demo" type="text/html">
            {{#
            for(var i=0;i<d.list.length;i++){ }}
            {{d.list[i]}}<br/>
            {{# } }}
        </script>

        <!--第二步：建立视图。用于呈现渲染结果。-->
        <div id="view"></div>

        <!--第三步：渲染模版-->
        <script>
            var laytpl = layui.laytpl;
            var getTpl = demo.innerHTML;
            var data = {
                list: ['a', 'b', 'c']
            };
            var view = document.getElementById('view');
            laytpl(getTpl).render(data, function(html) {
                view.innerHTML = html;
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.23 数据表格 table

注意：根据使用上的经验，建议不要把数据表格做为服务端分页的"前端复杂界面设计"的解决方案。当对数据表格进行自定义定制时，由于 Layui 框架中 table 模块可定制性不良的原因，也就是过度封装，有可能在复杂界面的实现上并不是太方便，还得自己添加 javascript 和 css 代码，建议使用普通 table 表格结合自写代码来实现分页，这样的高度可定制性能解决全部的界面设计及业务需求。当前端界面不需要自定义定制时，比如不需要更改对齐属性，不需要对 td 中的内容进行复杂的逻辑处理等情况，并且想快速实现分页效果时，数据表格还是首选的解决方案，完全可以胜任此类简单性的工作。

table 模块是 layui 2.0 版本中全新推出的，是 layui 最核心的组成之一，涵盖了日常业务所涉及的几乎全部需求。支持固定表头、固定行、固定列左/列右，支持拖拽改变列宽度，支持排序，支持多级表头，支持单元格的自定义模板，支持对表格重载（比如搜索、条件筛选等），支持复选框，支持分页，支持单元格编辑等等一些系列功能。table 模块是 layui 重点维护的项目之一。

模块加载名称：table。

1.23.1 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。

文件 new.html 代码如下：


```
{
    "code": 0,
    "msg": "",
    "count": 1000,
    "data": [{
        "id": 10000,
        "username": "user-0",
        "sex": "女",
        "city": "城市-0",
        "sign": "签名-0",
        "experience": 255,
        "logins": 24,
        "wealth": 82830700,
        "classify": "作家",
        "score": 57
    }, {
        "id": 10001,
        "username": "user-1",
        "sex": "男",
        "city": "城市-1",
```

```
        "sign": "签名-1",
        "experience": 884,
        "logins": 58,
        "wealth": 64928690,
        "classify": "词人",
        "score": 27
    }, {
        "id": 10002,
        "username": "user-2",
        "sex": "女",
        "city": "城市-2",
        "sign": "签名-2",
        "experience": 650,
        "logins": 77,
        "wealth": 6298078,
        "classify": "酱油",
        "score": 31
    }, {
        "id": 10003,
        "username": "user-3",
        "sex": "女",
        "city": "城市-3",
        "sign": "签名-3",
        "experience": 362,
        "logins": 157,
        "wealth": 37117017,
        "classify": "诗人",
        "score": 68
    }, {
        "id": 10004,
        "username": "user-4",
        "sex": "男",
        "city": "城市-4",
        "sign": "签名-4",
        "experience": 807,
        "logins": 51,
        "wealth": 76263262,
        "classify": "作家",
        "score": 6
    }, {
        "id": 10005,
        "username": "user-5",
        "sex": "女",
        "city": "城市-5",
```

```
        "sign": "签名-5",
        "experience": 173,
        "logins": 68,
        "wealth": 60344147,
        "classify": "作家",
        "score": 87
    }, {
        "id": 10006,
        "username": "user-6",
        "sex": "女",
        "city": "城市-6",
        "sign": "签名-6",
        "experience": 982,
        "logins": 37,
        "wealth": 57768166,
        "classify": "作家",
        "score": 34
    }, {
        "id": 10007,
        "username": "user-7",
        "sex": "男",
        "city": "城市-7",
        "sign": "签名-7",
        "experience": 727,
        "logins": 150,
        "wealth": 82030578,
        "classify": "作家",
        "score": 28
    }, {
        "id": 10008,
        "username": "user-8",
        "sex": "男",
        "city": "城市-8",
        "sign": "签名-8",
        "experience": 951,
        "logins": 133,
        "wealth": 16503371,
        "classify": "词人",
        "score": 14
    }, {
        "id": 10009,
        "username": "user-9",
        "sex": "女",
        "city": "城市-9",
```

```
        "sign": "签名-9",
        "experience": 484,
        "logins": 25,
        "wealth": 86801934,
        "classify": "词人",
        "score": 75
    }, {
        "id": 10010,
        "username": "user-10",
        "sex": "女",
        "city": "城市-10",
        "sign": "签名-10",
        "experience": 1016,
        "logins": 182,
        "wealth": 71294671,
        "classify": "诗人",
        "score": 34
    }, {
        "id": 10011,
        "username": "user-11",
        "sex": "女",
        "city": "城市-11",
        "sign": "签名-11",
        "experience": 492,
        "logins": 107,
        "wealth": 8062783,
        "classify": "诗人",
        "score": 6
    }, {
        "id": 10012,
        "username": "user-12",
        "sex": "女",
        "city": "城市-12",
        "sign": "签名-12",
        "experience": 106,
        "logins": 176,
        "wealth": 42622704,
        "classify": "词人",
        "score": 54
    }, {
        "id": 10013,
        "username": "user-13",
        "sex": "男",
        "city": "城市-13",
```

```
        "sign": "签名-13",
        "experience": 1047,
        "logins": 94,
        "wealth": 59508583,
        "classify": "诗人",
        "score": 63
    }, {
        "id": 10014,
        "username": "user-14",
        "sex": "男",
        "city": "城市-14",
        "sign": "签名-14",
        "experience": 873,
        "logins": 116,
        "wealth": 72549912,
        "classify": "词人",
        "score": 8
    }, {
        "id": 10015,
        "username": "user-15",
        "sex": "女",
        "city": "城市-15",
        "sign": "签名-15",
        "experience": 1068,
        "logins": 27,
        "wealth": 52737025,
        "classify": "作家",
        "score": 28
    }, {
        "id": 10016,
        "username": "user-16",
        "sex": "女",
        "city": "城市-16",
        "sign": "签名-16",
        "experience": 862,
        "logins": 168,
        "wealth": 37069775,
        "classify": "酱油",
        "score": 86
    }, {
        "id": 10017,
        "username": "user-17",
        "sex": "女",
        "city": "城市-17",
```

```
        "sign": "签名-17",
        "experience": 1060,
        "logins": 187,
        "wealth": 66099525,
        "classify": "作家",
        "score": 69
    }, {
        "id": 10018,
        "username": "user-18",
        "sex": "女",
        "city": "城市-18",
        "sign": "签名-18",
        "experience": 866,
        "logins": 88,
        "wealth": 81722326,
        "classify": "词人",
        "score": 74
    }, {
        "id": 10019,
        "username": "user-19",
        "sex": "女",
        "city": "城市-19",
        "sign": "签名-19",
        "experience": 682,
        "logins": 106,
        "wealth": 68647362,
        "classify": "词人",
        "score": 51
    }, {
        "id": 10020,
        "username": "user-20",
        "sex": "男",
        "city": "城市-20",
        "sign": "签名-20",
        "experience": 770,
        "logins": 24,
        "wealth": 92420248,
        "classify": "诗人",
        "score": 87
    }, {
        "id": 10021,
        "username": "user-21",
        "sex": "男",
        "city": "城市-21",
```

```
    "sign": "签名-21",
    "experience": 184,
    "logins": 131,
    "wealth": 71566045,
    "classify": "词人",
    "score": 99
}, {
    "id": 10022,
    "username": "user-22",
    "sex": "男",
    "city": "城市-22",
    "sign": "签名-22",
    "experience": 739,
    "logins": 152,
    "wealth": 60907929,
    "classify": "作家",
    "score": 18
}, {
    "id": 10023,
    "username": "user-23",
    "sex": "女",
    "city": "城市-23",
    "sign": "签名-23",
    "experience": 127,
    "logins": 82,
    "wealth": 14765943,
    "classify": "作家",
    "score": 30
}, {
    "id": 10024,
    "username": "user-24",
    "sex": "女",
    "city": "城市-24",
    "sign": "签名-24",
    "experience": 212,
    "logins": 133,
    "wealth": 59011052,
    "classify": "词人",
    "score": 76
}, {
    "id": 10025,
    "username": "user-25",
    "sex": "女",
    "city": "城市-25",
```

```
        "sign": "签名-25",
        "experience": 938,
        "logins": 182,
        "wealth": 91183097,
        "classify": "作家",
        "score": 69
    }, {
        "id": 10026,
        "username": "user-26",
        "sex": "男",
        "city": "城市-26",
        "sign": "签名-26",
        "experience": 978,
        "logins": 7,
        "wealth": 48008413,
        "classify": "作家",
        "score": 65
    }, {
        "id": 10027,
        "username": "user-27",
        "sex": "女",
        "city": "城市-27",
        "sign": "签名-27",
        "experience": 371,
        "logins": 44,
        "wealth": 64419691,
        "classify": "诗人",
        "score": 60
    }, {
        "id": 10028,
        "username": "user-28",
        "sex": "女",
        "city": "城市-28",
        "sign": "签名-28",
        "experience": 977,
        "logins": 21,
        "wealth": 75935022,
        "classify": "作家",
        "score": 37
    }, {
        "id": 10029,
        "username": "user-29",
        "sex": "男",
        "city": "城市-29",
```

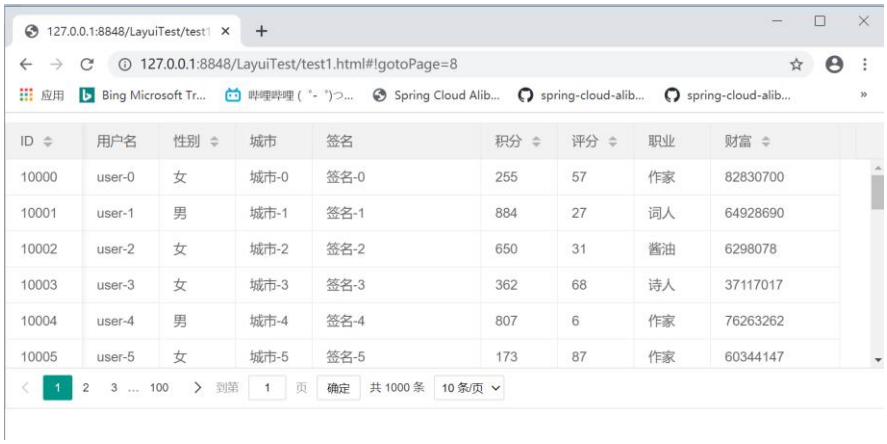
```
        "sign": "签名-29",
        "experience": 647,
        "logins": 107,
        "wealth": 97450636,
        "classify": "酱油",
        "score": 27
    }]
}
```

运行文件代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>

        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                height: 300,
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            width: 80,
                            sort: true,
                            fixed: 'left'
                        }, {
                            field: 'username',
                            title: '用户名',
                            width: 80
                        }, {
                            field: 'sex',
```

```
                    title: '性别',
                    width: 80,
                    sort: true
                }, {
                    field: 'city',
                    title: '城市',
                    width: 80
                }, {
                    field: 'sign',
                    title: '签名',
                    width: 177
                }, {
                    field: 'experience',
                    title: '积分',
                    width: 80,
                    sort: true
                }, {
                    field: 'score',
                    title: '评分',
                    width: 80,
                    sort: true
                }, {
                    field: 'classify',
                    title: '职业',
                    width: 80
                }, {
                    field: 'wealth',
                    title: '财富',
                    width: 135,
                    sort: true
                }
            ]
        ]
    });
</script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.23.2 三种初始化渲染方式

table 模块做了三种初始化的支持，可按照个人喜好和实际情况灵活使用，如图 1-xx 所示。

	方式	机制	适用场景
01.	方法渲染	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	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	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

1.23.2.1 方法渲染

“方法渲染”是将基础参数的设置放在了 JS 代码中，且原始的 table 标签只需要一个选择器。

文件 new.html 测试代码如下：

```
{
    "code": 0,
    "msg": "",
    "count": 1000,
    "data": [{
        "id": 10000,
        "username": "username1",
        "password": "password1"
    },
    {
        "id": 20000,
        "username": "username2",
        "password": "password2"
    },
```

```
    {
        "id": 30000,
        "username": "username3",
        "password": "password3"
    }
    ]
}
```

运行文件代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            width: 200,
                            sort: true,
                            fixed: 'left'
                        }, {
                            field: 'username',
                            title: '账号',
                            width: 200
                        }, {
                            field: 'password',
                            title: '账号',
                            width: 200,
                            sort: true
```

```
                    }
                ]
            ]
        });
    </script>
</body>
</html>
```

elem 值类型 String/DOM，指定原始 table 容器的选择器或 DOM 对象，是方法渲染方式的必填项。

cols 值类型 Array，设置表头，值是一个二维数组，是方法渲染方式的必填项。

运行效果如图 1-xx 所示。

推荐采用“方法渲染”，其最大的优势在于可以脱离 HTML 文件而专注于 JS 本身，尤其对于项目的频繁改动及发布，其便捷性会体现得更为明显。

table.render()方法返回一个 table 对象：

var tableObject = table.render(options);

可用于对当前表格进行“重加载 reload”等操作，关于“重加载 reload”请参看后面的章节。

1.23.2.2 自动渲染

所谓的自动渲染是在一段 table 容器中配置好相应的参数，由 table 模块内部自动对其完成渲染，而无需写初始的渲染方法。

需要关注的是以下三点：

（1）带有 class="layui-table"的<table>标签。

（2）对标签设置属性 lay-data=""，用于配置一些基础参数。

（3）在<th>标签中设置属性 lay-data=""用于配置表头信息。

按照上述的规范写好 table 原始容器后，只要加载了 layui 的 table 模块，就会自动对其建立动态的数据表格。

文件 new.html 代码如下：

```
{
    "code": 0,
    "msg": "",
    "count": 4,
```

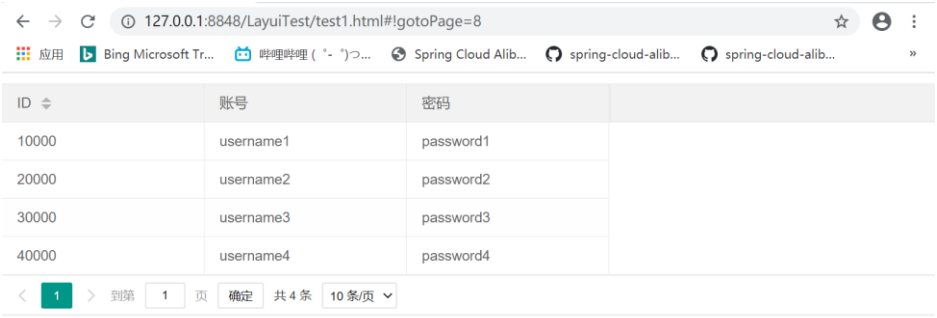
```
    "data": [{
        "id": 10000,
        "username": "username1",
        "password": "password1"
    },
    {
        "id": 20000,
        "username": "username2",
        "password": "password2"
    },
    {
        "id": 30000,
        "username": "username3",
        "password": "password3"
    },
    {
        "id": 40000,
        "username": "username4",
        "password": "password4"
    }
    ]
}
```

运行文件代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />

    </head>
    <body>
        <table class="layui-table" lay-data="{url:'new.html', page:true,
id:'test'}" lay-filter="test">
            <thead>
                <tr>
                    <th lay-data="{field:'id', width:200, sort:
true}">ID</th>
                    <th lay-data="{field:'username', width:200}">账号</th>
                    <th lay-data="{field:'password', width:200}">密码</th>
                </tr>
            </thead>
```

```
        </table>
        <script src="layui/layui.all.js"></script>
        <script>
            var table = layui.table;
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.23.2.3 转换静态表格

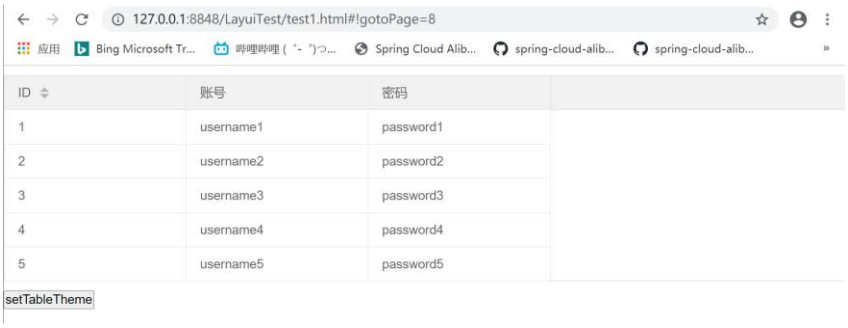
假设页面已经存在有内容的表格，它由原始的 table 标签组成，这时需要赋予它一些扩展的属性，比如拖拽调整列宽，比如排序等等功能，那么同样可以很轻松地去产生数据表格。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />

    </head>
    <body>
        <table lay-filter="test">
            <thead>
                <tr>
                    <th lay-data="{field:'id', width:200, sort:
true}">ID</th>
                    <th lay-data="{field:'username', width:200}">账号</th>
                    <th lay-data="{field:'password', width:200}">密码</th>
                </tr>
            </thead>
```

```
            <tr>
                <td>1</td>
                <td>username1</td>
                <td>password1</td>
            </tr>
            <tr>
                <td>2</td>
                <td>username2</td>
                <td>password2</td>
            </tr>
            <tr>
                <td>3</td>
                <td>username3</td>
                <td>password3</td>
            </tr>
            <tr>
                <td>4</td>
                <td>username4</td>
                <td>password4</td>
            </tr>
            <tr>
                <td>5</td>
                <td>username5</td>
                <td>password5</td>
            </tr>
        </table>
        <input type="button" value="setTableTheme" id="setTableTheme">
        <script src="layui/layui.all.js"></script>
        <script>
            $(document).ready(function() {
                $("#setTableTheme").click(function() {
                    var table = layui.table;
                    //转换静态表格
                    table.init('test', {});
                });
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.23.3 基础参数一览表

基础参数并非所有都要出现，有必选也有可选，结合实际需求自由设置即可。基础参数一般出现在以下三种场景中。

（1）场景一是在方法渲染中使用，如图 1-xx 所示。

```
场景一：下述方法中的键值即为基础参数项
table.render({
  height: 300
  ,url: '/api/data'
});
```

（2）场景二是在自动渲染中使用，如图 1-xx 所示。

```
场景二：下述 lay-data 里面的内容即为基础参数项，切记：值要用单引号
<table lay-data="{height:300, url:'/api/data'}" lay-filter="demo"> …… </table>
```

（3）场景三是在重加载 reload 中使用，如图 1-xx 所示。

```
更多场景：下述 options 即为含有基础参数项的对象
> table.init('filter', options); //转化静态表格
> var tableObj = table.render({});
  tableObj.reload(options); //重载表格
```

图 1-xx 所示是目前 table 模块所支持的全部参数一览表，并对重点参数进行了详细说明。

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
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
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: ' xxx ' *//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	该参数可自由配置头部工具栏右侧的图标按钮	详见头工具栏图标
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
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
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 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。 注意： *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]。 注意： *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
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
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
id	String	设定容器唯一 id。id 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，你在下文诸多地方都将会见识它的存在。 值得注意的是：从 layui 2.2.0 开始，该参数也可以自动从 *<table id="test"></table>* 中的 id 参数中获取。	test
skin（等）	-	设定表格各种外观、尺寸等	详见表格风格

1.23.3.1 toolbar 参数

值类型：String/DOM/Boolean。

开启表格头部工具栏区域，该参数支持四种类型值：

（1）toolbar:'#toolbarDemo' //指向自定义工具栏模板选择器

（2）toolbar: ' xxx ' //直接传入工具栏模板 html 代码

（3）toolbar: true //仅开启工具栏，不显示左侧模板，若需要“列显示隐藏”、“导出”、“打印”等功能，则必须开启该参数

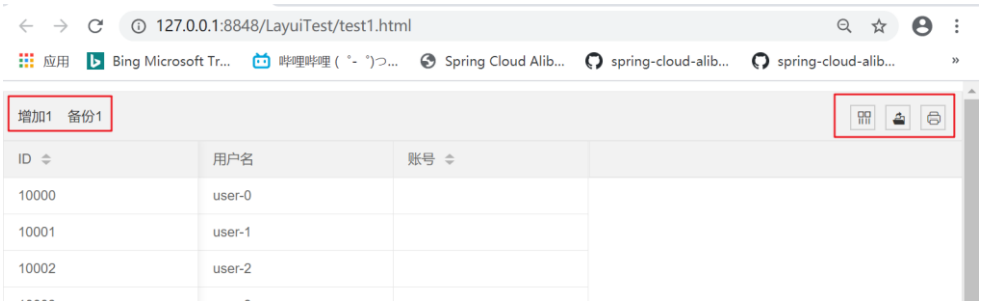
（4）toolbar: 'default' //让工具栏左侧显示默认的内置模板

1.23.3.1.1 toolbar: '#toolbarDemo'

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />

    </head>
    <body>
        <div id="toolbarDemo" style="display: none;">
            <a href='#'>增加1</a>
              
            <a href='#'>备份1</a>
        </div>
        <table id="demo" lay-filter="test"></table>
        <script src="layui/layui.all.js"></script>
        <script>
            layui.use('table', function() {
                var table = layui.table;
                //第一个实例
                table.render({
                    elem: '#demo',
                    url: 'new.html', //数据接口
                    page: true, //开启分页
                    toolbar: "#toolbarDemo",
                    cols: [
                        [ //表头
                            {
                                field: 'id',
                                title: 'ID',
                                width: 200,
```

```
                        sort: true,
                        fixed: 'left'
                    }, {
                        field: 'username',
                        title: '用户名',
                        width: 200
                    }, {
                        field: 'password',
                        title: '账号',
                        width: 200,
                        sort: true
                    }
                ]
            ]
        });
    });
</script>
</body>
</html>
```

运行效果如图 1-xx 所示。

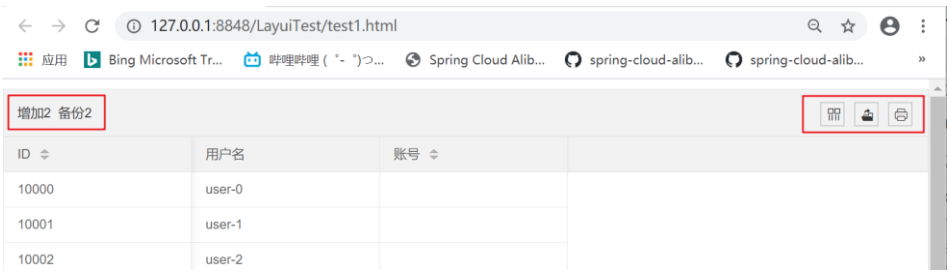


1.23.3.1.2 toolbar: ' xxx '

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />

    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
```

```
        <script src="layui/layui.all.js"></script>
        <script>
            layui.use('table', function() {
                var table = layui.table;
                //第一个实例
                table.render({
                    elem: '#demo',
                    url: 'new.html', //数据接口
                    page: true, //开启分页
                    toolbar: '<div style="display: none;"><a href="#">增加
2</a>  <a href="#">备份2</a></div>',
                    cols: [
                        [ //表头
                            {
                                field: 'id',
                                title: 'ID',
                                width: 200,
                                sort: true,
                                fixed: 'left'
                            }, {
                                field: 'username',
                                title: '用户名',
                                width: 200
                            }, {
                                field: 'password',
                                title: '账号',
                                width: 200,
                                sort: true
                            }
                        ]
                    ]
                });
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

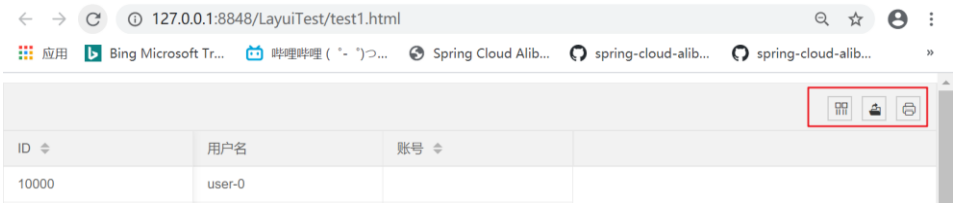


1.23.3.1.3 toolbar: true

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />

    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script src="layui/layui.all.js"></script>
        <script>
            layui.use('table', function() {
                var table = layui.table;
                //第一个实例
                table.render({
                    elem: '#demo',
                    url: 'new.html', //数据接口
                    page: true, //开启分页
                    toolbar: true,
                    cols: [
                        [ //表头
                            {
                                field: 'id',
                                title: 'ID',
                                width: 200,
                                sort: true,
                                fixed: 'left'
                            }, {
                                field: 'username',
                                title: '用户名',
                                width: 200
                            }, {
                                field: 'password',
                                title: '账号',
                                width: 200,
                                sort: true
                            }
                        ]
                    ]
                });
```

```
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

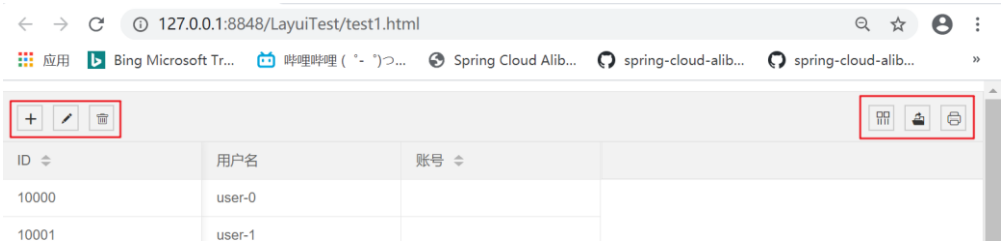


1.23.3.1.4 toolbar: 'default'

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script src="layui/layui.all.js"></script>
        <script>
            layui.use('table', function() {
                var table = layui.table;
                //第一个实例
                table.render({
                    elem: '#demo',
                    url: 'new.html', //数据接口
                    page: true, //开启分页
                    toolbar: 'default',
                    cols: [
                        [ //表头
                            {
                                field: 'id',
                                title: 'ID',
                                width: 200,
                                sort: true,
                                fixed: 'left'
                            }, {
                                field: 'username',
```

```
                    title: '用户名',
                    width: 200
                }, {
                    field: 'password',
                    title: '账号',
                    width: 200,
                    sort: true
                }
            ]
        ]
    });
});
</script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.23.3.2 defaultToolbar

defaultToolbar 值类型 Array，默认值：["filter","exports","print"]。该参数可自由配置头部工具栏右侧的图标按钮。

Array 数组支持以下可选值：

（1）filter：显示筛选图标

（2）exports：显示导出图标

（3）print：显示打印图标

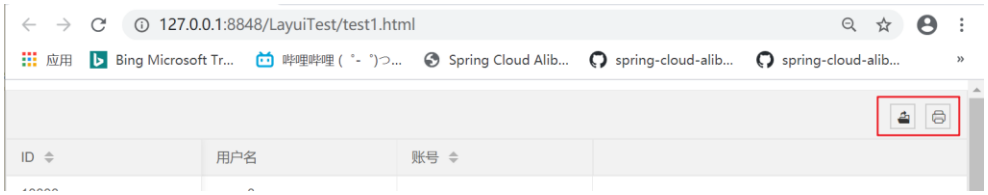
可根据值的顺序显示排版图标，如：

defaultToolbar: ['filter', 'print', 'exports']

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
```

```
    <body>
        <table id="demo" lay-filter="test"></table>

        <script>
            layui.use('table', function() {
                var table = layui.table;
                //第一个实例
                table.render({
                    elem: '#demo',
                    url: 'new.html', //数据接口
                    page: true, //开启分页
                    toolbar: true,
                    defaultToolbar: ["exports", "print"],
                    cols: [
                        [ //表头
                            {
                                field: 'id',
                                title: 'ID',
                                width: 200,
                                sort: true,
                                fixed: 'left'
                            }, {
                                field: 'username',
                                title: '用户名',
                                width: 200
                            }, {
                                field: 'password',
                                title: '账号',
                                width: 200,
                                sort: true
                            }
                        ]
                    ]
                });
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.23.3.3 width

width 值类型 Number，作用设定容器宽度。table 容器的默认宽度是跟随它的父元素同宽，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。

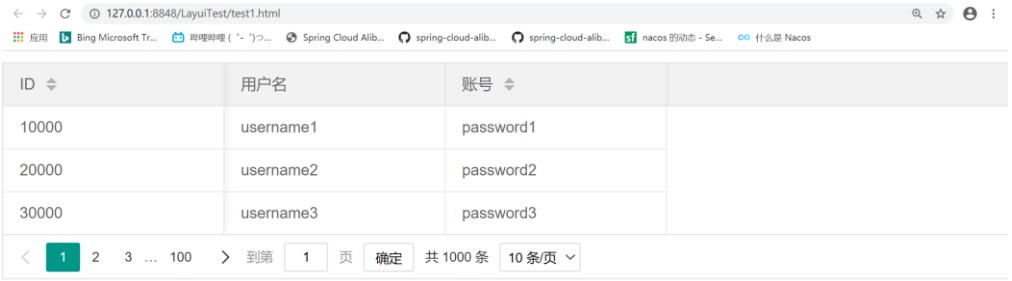
1.23.3.3.1 不添写

宽度自适应的测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>

        <script>
            layui.use('table', function() {
                var table = layui.table;
                //第一个实例
                table.render({
                    elem: '#demo',
                    url: 'new.html', //数据接口
                    page: true, //开启分页
                    cols: [
                        [ //表头
                            {
                                field: 'id',
                                title: 'ID',
                                width: 200,
                                sort: true,
                                fixed: 'left'
                            }, {
                                field: 'username',
```

```
                    title: '用户名',
                    width: 200
                }, {
                    field: 'password',
                    title: '账号',
                    width: 200,
                    sort: true
                }
            ]
        ]
    });
});
</script>
</body>
</html>
```

运行效果如图 1-xx 所示。



表格的宽度和浏览器宽度一样。

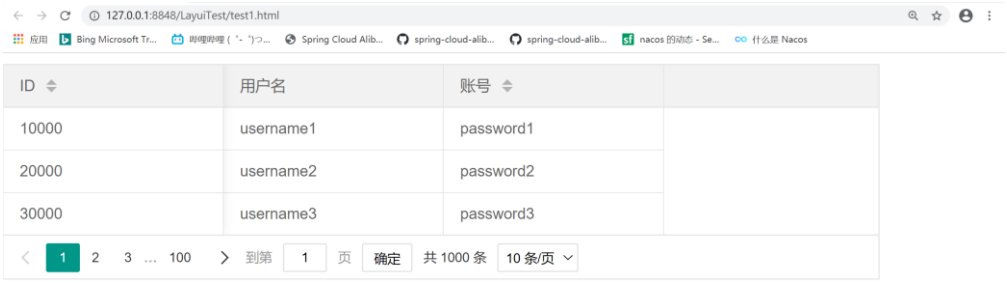
1.23.3.3.2 固定值

指定宽度的测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>

        <script>
            layui.use('table', function() {
```

```
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                width: 800,
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            width: 200,
                            sort: true,
                            fixed: 'left'
                        }, {
                            field: 'username',
                            title: '用户名',
                            width: 200
                        }, {
                            field: 'password',
                            title: '账号',
                            width: 200,
                            sort: true
                        }
                    ]
                ]
            });
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



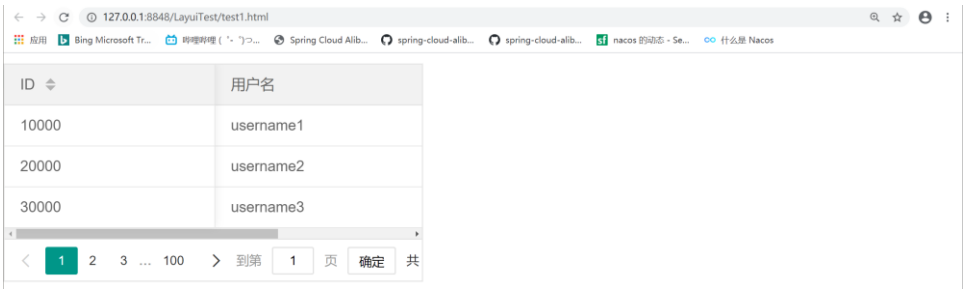
1.23.3.3.3 出现水平滚动条

出现水平滚动条的测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>

        <script>
            layui.use('table', function() {
                var table = layui.table;
                //第一个实例
                table.render({
                    elem: '#demo',
                    url: 'new.html', //数据接口
                    page: true, //开启分页
                    width: 400,
                    cols: [
                        [ //表头
                            {
                                field: 'id',
                                title: 'ID',
                                width: 200,
                                sort: true,
                                fixed: 'left'
                            }, {
                                field: 'username',
                                title: '用户名',
                                width: 200
                            }, {
                                field: 'password',
                                title: '账号',
                                width: 200,
                                sort: true
                            }
                        ]
                    ]
```

```
            });
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.23.3.4 height

height 值类型 Number/String，设定容器高度。
可选值如图 1-xx 所示。

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'

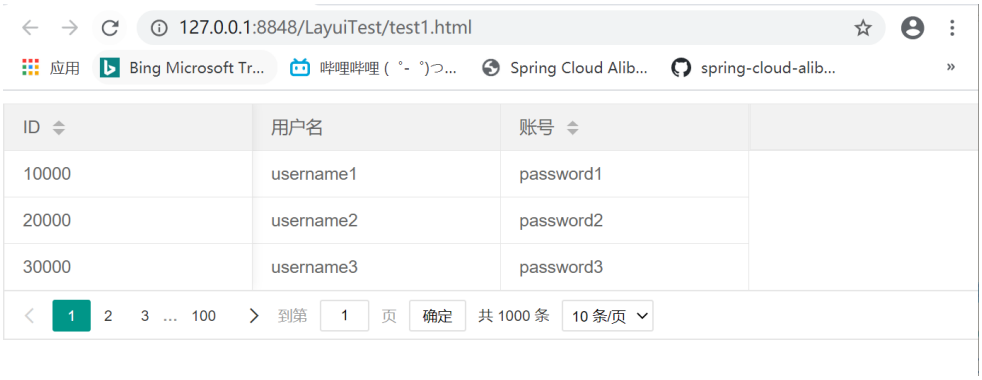
1.23.3.4.1 不添写

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>

        <script>
            layui.use('table', function() {
```

```
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            width: 200,
                            sort: true,
                            fixed: 'left'
                        }, {
                            field: 'username',
                            title: '用户名',
                            width: 200
                        }, {
                            field: 'password',
                            title: '账号',
                            width: 200,
                            sort: true
                        }
                    ]
                ]
            });
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



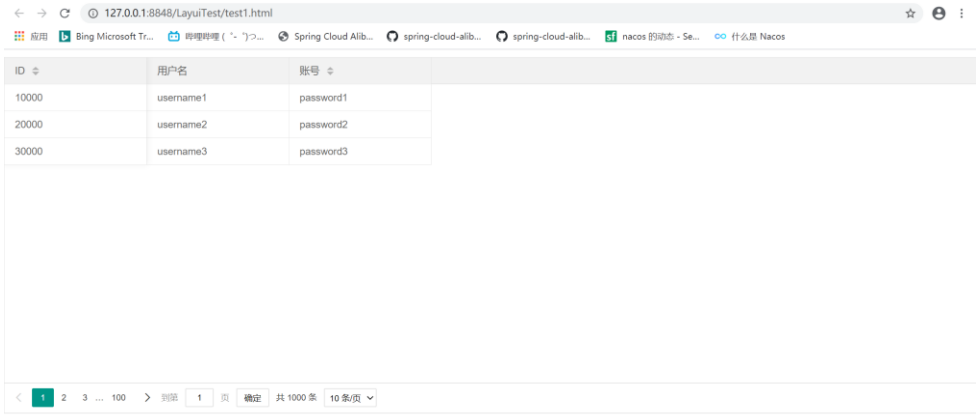
1.23.3.4.2 固定值

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>

        <script>
            layui.use('table', function() {
                var table = layui.table;
                //第一个实例
                table.render({
                    elem: '#demo',
                    url: 'new.html', //数据接口
                    page: true, //开启分页
                    height: 500,
                    cols: [
                        [ //表头
                            {
                                field: 'id',
                                title: 'ID',
                                width: 200,
                                sort: true,
                                fixed: 'left'
                            }, {
                                field: 'username',
                                title: '用户名',
                                width: 200
                            }, {
                                field: 'password',
                                title: '账号',
                                width: 200,
                                sort: true
                            }
                        ]
                    ]
```

```
            });
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。

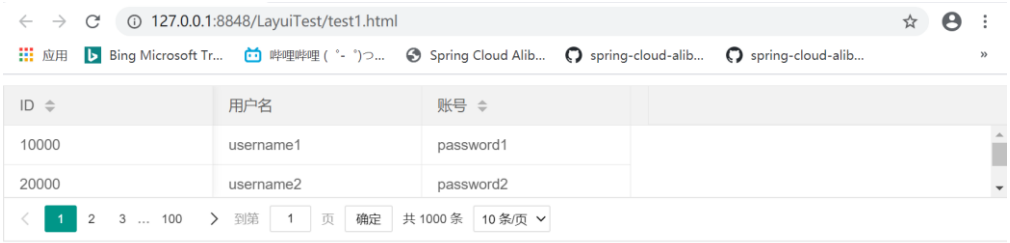


出现垂直滚动条的测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>

        <script>
            layui.use('table', function() {
                var table = layui.table;
                //第一个实例
                table.render({
                    elem: '#demo',
                    url: 'new.html', //数据接口
                    page: true, //开启分页
                    height: 150,
                    cols: [
                        [ //表头
                            {
```

```
                    field: 'id',
                    title: 'ID',
                    width: 200,
                    sort: true,
                    fixed: 'left'
                }, {
                    field: 'username',
                    title: '用户名',
                    width: 200
                }, {
                    field: 'password',
                    title: '账号',
                    width: 200,
                    sort: true
                }
            ]
        ]
    });
});
</script>
</body>
</html>
```

运行效果如图 1-xx 所示。



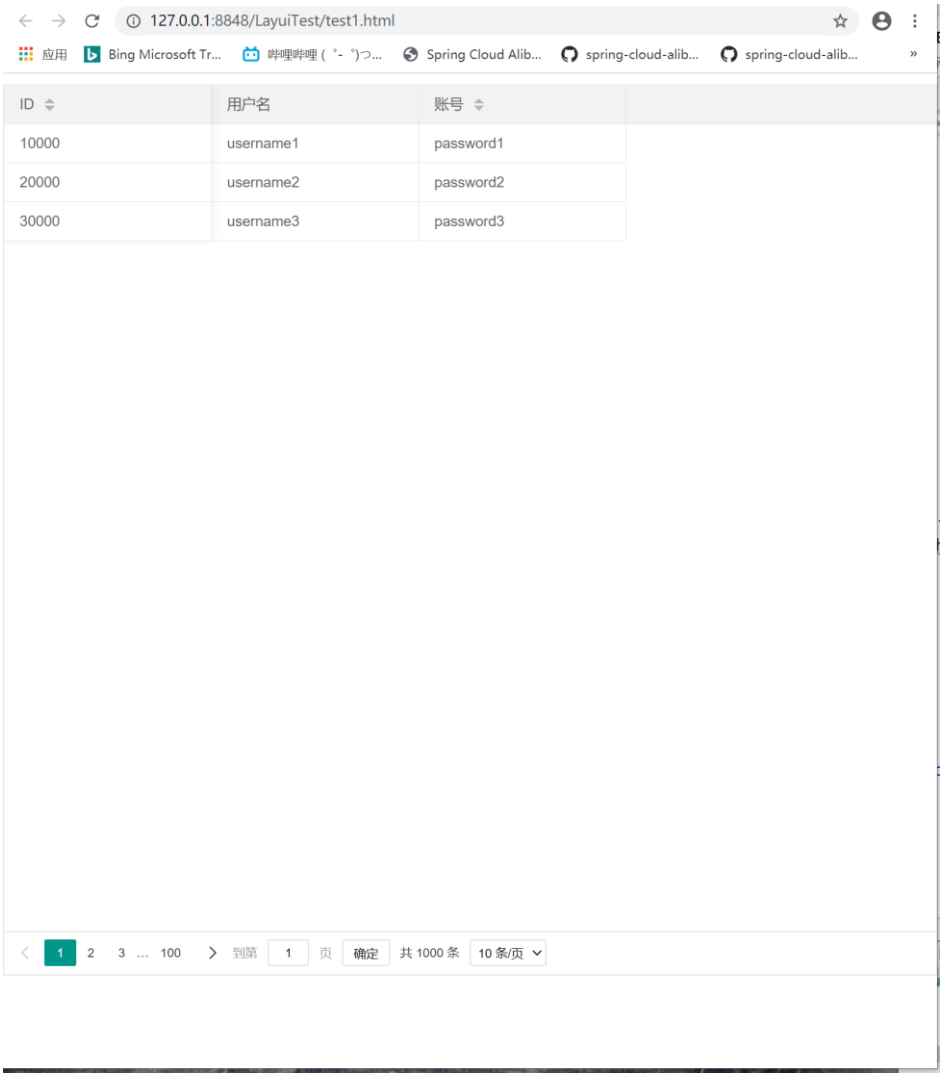
1.23.3.4.3 full-差值

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
```

```
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            layui.use('table', function() {
                var table = layui.table;
                //第一个实例
                table.render({
                    elem: '#demo',
                    url: 'new.html', //数据接口
                    page: true, //开启分页
                    height: "full-100",
                    cols: [
                        [ //表头
                            {
                                field: 'id',
                                title: 'ID',
                                width: 200,
                                sort: true,
                                fixed: 'left'
                            }, {
                                field: 'username',
                                title: '用户名',
                                width: 200
                            }, {
                                field: 'password',
                                title: '账号',
                                width: 200,
                                sort: true
                            }
                        ]
                    ]
                });
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.23.3.5 cellMinWidth

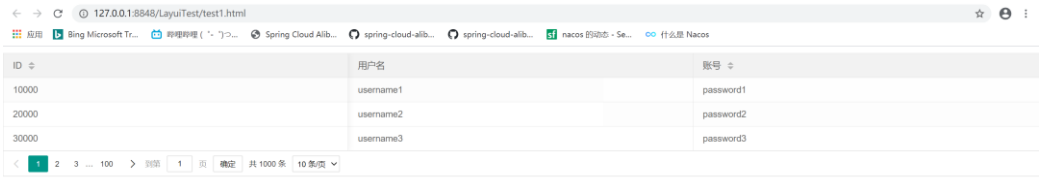
cellMinWidth 值类型 Number。全局定义所有常规单元格的最小宽度（默认：60），一般用于列宽自动分配的情况，其优先级低于表头参数中的 minWidth。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
```

```
        <script>
            layui.use('table', function() {
                var table = layui.table;
                //第一个实例
                table.render({
                    elem: '#demo',
                    url: 'new.html', //数据接口
                    page: true, //开启分页
                    cellMinWidth: 400,
                    cols: [
                        [ //表头
                            {
                                field: 'id',
                                title: 'ID',
                                sort: true,
                                fixed: 'left'
                            }, {
                                field: 'username',
                                title: '用户名',
                            }, {
                                field: 'password',
                                title: '账号',
                                sort: true
                            }
                        ]
                    ]
                });
            });
        </script>
    </body>
</html>
```

浏览器宽度较大的运行效果如图 1-xx 所示。



浏览器宽度较小的运行效果如图 1-xx 所示。

1.23.3.6 done

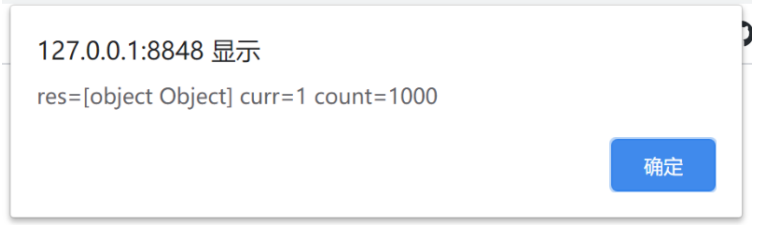
done 值类型 Function。无论是异步请求数据，还是直接赋值数据，都会触发该回调。可以利用该回调做一些表格以外元素的渲染。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>

        <script>
            layui.use('table', function() {
                var table = layui.table;
                //第一个实例
                table.render({
                    elem: '#demo',
                    url: 'new.html', //数据接口
                    page: true, //开启分页
                    done: function(res, curr, count) {
                        alert("res=" + res + " curr=" + curr + " count=" +
count);

                    },
                    cols: [
                        [ //表头
                            {
                                field: 'id',
```

```
                        title: 'ID',
                        sort: true,
                        fixed: 'left'
                    }, {
                        field: 'username',
                        title: '用户名',
                    }, {
                        field: 'password',
                        title: '账号',
                        sort: true
                    }
                ]
            ]
        });
    });
</script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.23.3.7 data

data 值类型 Array。直接赋值数据，既适用于只展示一页数据，也适用于展示多页数据。

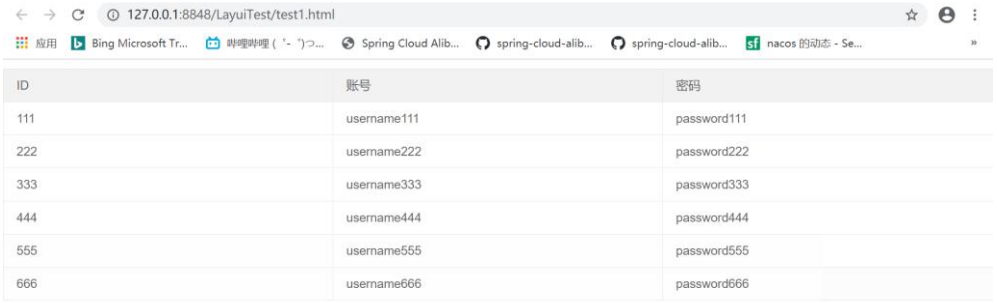
属性 data 的基本使用，测试代码如下：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title></title>
    <script src="js/jquery3.5.1.js"></script>
    <link rel="stylesheet" href="layui/css/layui.css" />
</head>
<body>
    <table id="demo" lay-filter="test"></table>
    <script src="layui/layui.all.js"></script>
```

```
<script>
    var table = layui.table;
    //第一个实例
    table.render({
        elem: "#demo",
        cols: [
            [ //表头
                {
                    field: 'id',
                    title: 'ID',
                }, {
                    field: 'username',
                    title: '账号'
                }, {
                    field: 'password',
                    title: '密码',
                }
            ]
        ],
        data: [{
            "id": 111,
            "username": "username111",
            "password": "password111"
        }, {
            "id": 222,
            "username": "username222",
            "password": "password222"
        }, {
            "id": 333,
            "username": "username333",
            "password": "password333"
        }, {
            "id": 444,
            "username": "username444",
            "password": "password444"
        }, {
            "id": 555,
            "username": "username555",
            "password": "password555"
        }, {
            "id": 666,
            "username": "username666",
            "password": "password666"
        }]
```

```
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

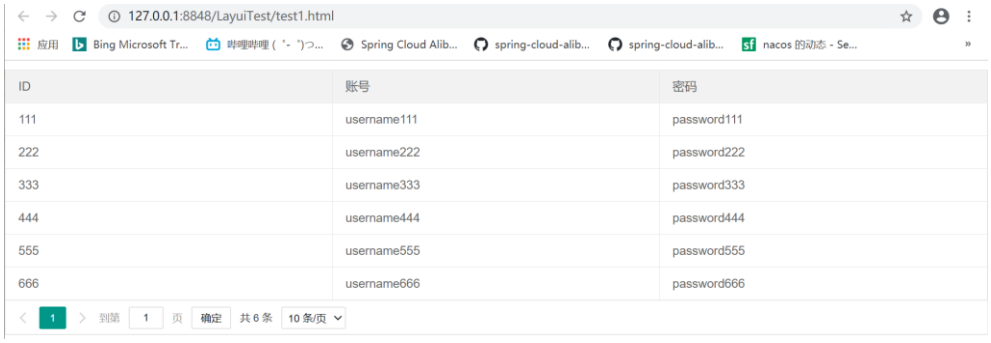


开启分页，测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />

    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script src="layui/layui.all.js"></script>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: "#demo",
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                        }, {
                            field: 'username',
                            title: '账号'
                        }, {
                            field: 'password',
```

```
                    title: '密码',
                }
            ]
        ],
        data: [{
            "id": 111,
            "username": "username111",
            "password": "password111"
        }, {
            "id": 222,
            "username": "username222",
            "password": "password222"
        }, {
            "id": 333,
            "username": "username333",
            "password": "password333"
        }, {
            "id": 444,
            "username": "username444",
            "password": "password444"
        }, {
            "id": 555,
            "username": "username555",
            "password": "password555"
        }, {
            "id": 666,
            "username": "username666",
            "password": "password666"
        }]
    });
</script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.23.3.8 totalRow

totalRow 值类型 Boolean。是否开启合计行区域。

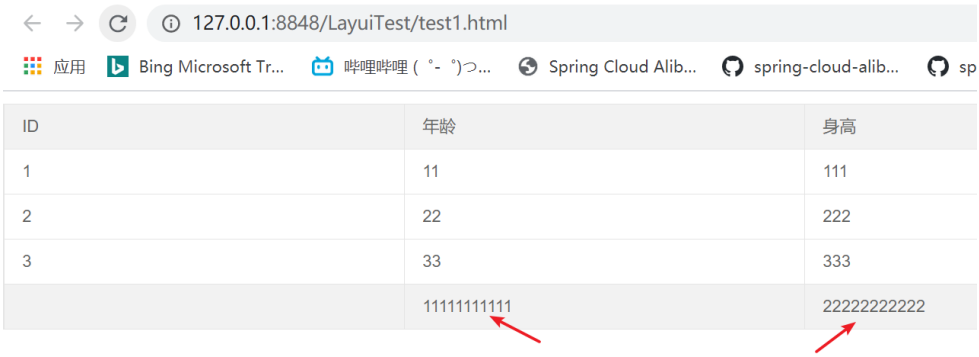
注意：需要结合 cols 表头参数 totalRowText 或服务端返回数据的 totalRow 属性进行使用。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />

    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script src="layui/layui.all.js"></script>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: "#demo",
                totalRow: true,
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID'
                        }, {
                            field: 'age',
                            title: '年龄',
                            totalRowText: '11111111111'
                        }, {
                            field: 'height',
                            title: '身高',
                            totalRowText: '22222222222'
                        }
                    ]
                ],
                data: [{
```

```
                "id": 1,
                "age": 11,
                "height": 111
            }, {
                "id": 2,
                "age": 22,
                "height": 222
            }, {
                "id": 3,
                "age": 33,
                "height": 333
            }]
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.23.3.9 page

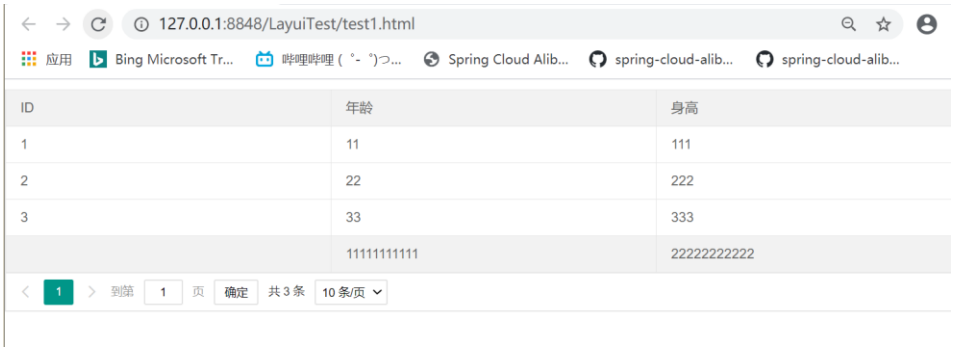
page 值类型 Boolean/Object。开启分页（默认：false）。支持传入一个对象，里面可包含 laypage 组件所有支持的参数（jump、elem 除外）。

开启分页的测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />

    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
```

```
        <script src="layui/layui.all.js"></script>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: "#demo",
                page: true, //开启分页
                totalRow: true,
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID'
                        }, {
                            field: 'age',
                            title: '年龄',
                            totalRowText: '11111111111'
                        }, {
                            field: 'height',
                            title: '身高',
                            totalRowText: '22222222222'
                        }
                    ]
                ],
                data: [{
                    "id": 1,
                    "age": 11,
                    "height": 111
                }, {
                    "id": 2,
                    "age": 22,
                    "height": 222
                }, {
                    "id": 3,
                    "age": 33,
                    "height": 333
                }]
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.23.3.10 limit

limit 值类型 Number。每页显示的条数（默认：10）。该值一定要对应 limits 参数。注意：优先级低于 page 参数中的 limit 参数。

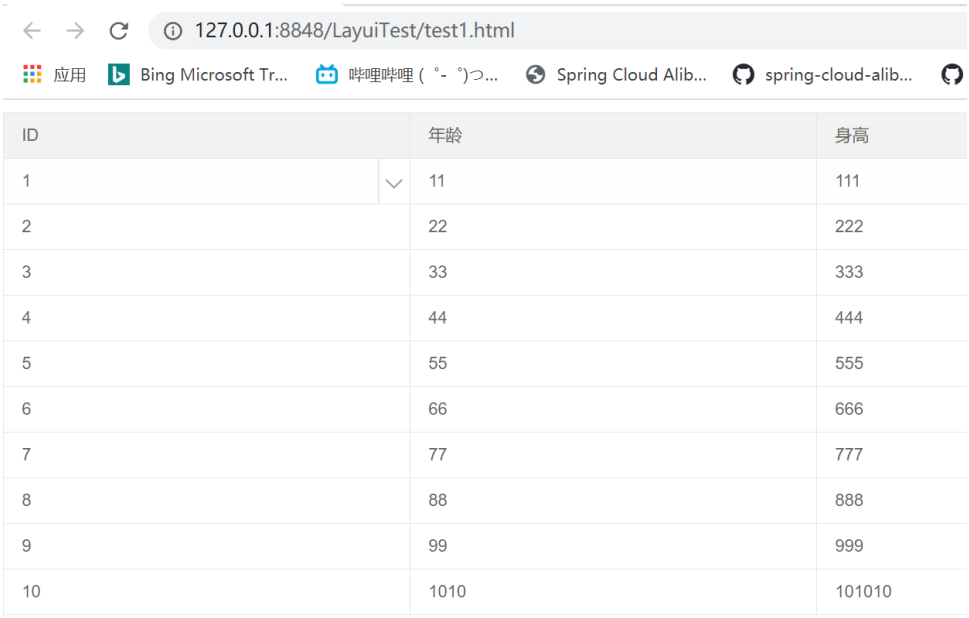
测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />

    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script src="layui/layui.all.js"></script>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: "#demo",
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID'
                        }, {
                            field: 'age',
                            title: '年龄'
                        }, {
                            field: 'height',
                            title: '身高'
```

```
        }
    ]
],
data: [{
    "id": 1,
    "age": 11,
    "height": 111
}, {
    "id": 2,
    "age": 22,
    "height": 222
}, {
    "id": 3,
    "age": 33,
    "height": 333
}, {
    "id": 4,
    "age": 44,
    "height": 444
}, {
    "id": 5,
    "age": 55,
    "height": 555
}, {
    "id": 6,
    "age": 66,
    "height": 666
}, {
    "id": 7,
    "age": 77,
    "height": 777
}, {
    "id": 8,
    "age": 88,
    "height": 888
}, {
    "id": 9,
    "age": 99,
    "height": 999
}, {
    "id": 10,
    "age": 1010,
    "height": 101010
}, {
```

```
                "id": 11,
                "age": 1111,
                "height": 111111
            }, {
                "id": 12,
                "age": 1212,
                "height": 121212
            }]
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



默认一页显示 10 条记录。

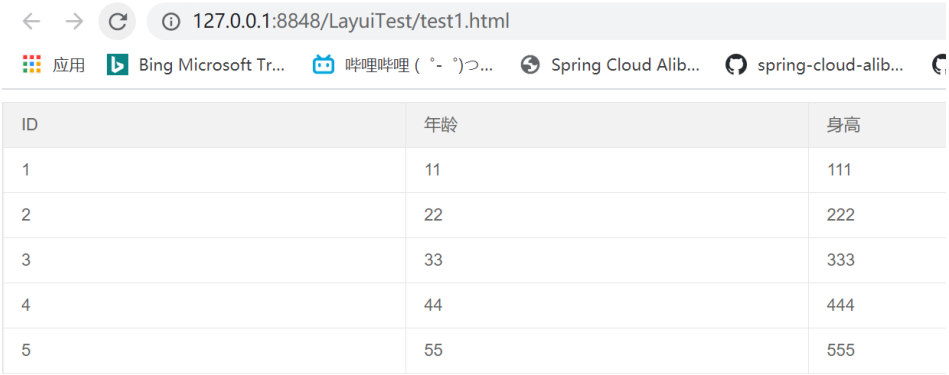
一页显示 5 条记录的测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />

    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
```

```
<script src="layui/layui.all.js"></script>
<script>
    var table = layui.table;
    //第一个实例
    table.render({
        elem: "#demo",
        limit: 5,
        cols: [
            [ //表头
                {
                    field: 'id',
                    title: 'ID'
                }, {
                    field: 'age',
                    title: '年龄'
                }, {
                    field: 'height',
                    title: '身高'
                }
            ]
        ],
        data: [{
            "id": 1,
            "age": 11,
            "height": 111
        }, {
            "id": 2,
            "age": 22,
            "height": 222
        }, {
            "id": 3,
            "age": 33,
            "height": 333
        }, {
            "id": 4,
            "age": 44,
            "height": 444
        }, {
            "id": 5,
            "age": 55,
            "height": 555
        }, {
            "id": 6,
            "age": 66,
```

```
                "height": 666
            }, {
                "id": 7,
                "age": 77,
                "height": 777
            }, {
                "id": 8,
                "age": 88,
                "height": 888
            }, {
                "id": 9,
                "age": 99,
                "height": 999
            }, {
                "id": 10,
                "age": 1010,
                "height": 101010
            }, {
                "id": 11,
                "age": 1111,
                "height": 111111
            }, {
                "id": 12,
                "age": 1212,
                "height": 121212
            }]
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.23.3.11 limits

limits 值类型 Array。每页条数的选择项，默认：[10,20,30,40,50,60,70,80,90]。
注意：优先级低于 page 参数中的 limits 参数。

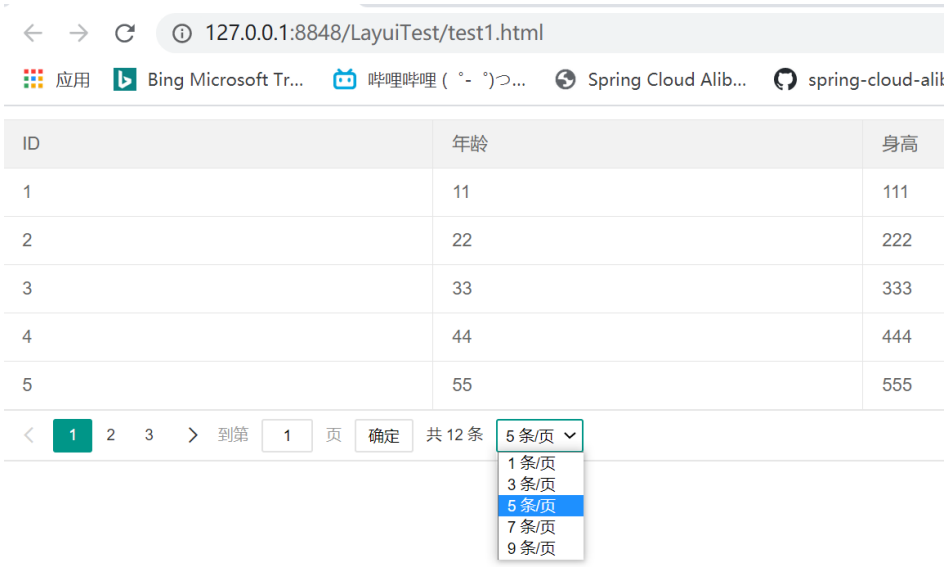
测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />

    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script src="layui/layui.all.js"></script>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: "#demo",
                page: true,
                limit: 5,
                limits: [1, 3, 5, 7, 9],
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID'
                        }, {
                            field: 'age',
                            title: '年龄'
                        }, {
                            field: 'height',
                            title: '身高'
                        }
                    ]
                ],
                data: [{
                    "id": 1,
                    "age": 11,
                    "height": 111
```

```
}, {
    "id": 2,
    "age": 22,
    "height": 222
}, {
    "id": 3,
    "age": 33,
    "height": 333
}, {
    "id": 4,
    "age": 44,
    "height": 444
}, {
    "id": 5,
    "age": 55,
    "height": 555
}, {
    "id": 6,
    "age": 66,
    "height": 666
}, {
    "id": 7,
    "age": 77,
    "height": 777
}, {
    "id": 8,
    "age": 88,
    "height": 888
}, {
    "id": 9,
    "age": 99,
    "height": 999
}, {
    "id": 10,
    "age": 1010,
    "height": 101010
}, {
    "id": 11,
    "age": 1111,
    "height": 111111
}, {
    "id": 12,
    "age": 1212,
    "height": 121212
```

```
            }]
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.23.3.12 loading

loading 值类型 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。

1.23.3.13 title

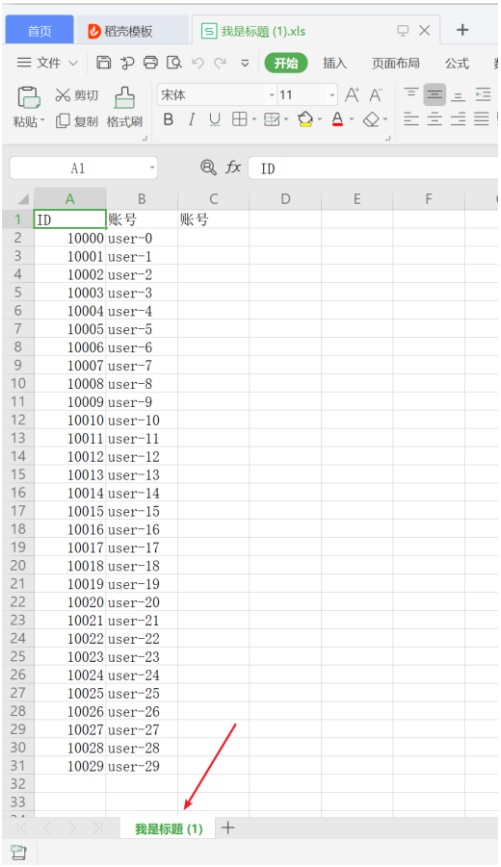
title 值类型 String。定义 table 的大标题（在文件导出等地方会用到，对应 Excel 标签名）。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
```

```
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                toolbar: true,
                title: "我是标题",
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            width: 200,
                            sort: true,
                            fixed: 'left'
                        }, {
                            field: 'username',
                            title: '账号',
                            width: 200
                        }, {
                            field: 'password',
                            title: '账号',
                            width: 200,
                            sort: true
                        }
                    ]
                ]
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.23.3.14 text

text 值类型 Object。自定义文本，如空数据时显示的异常提示。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                page: true, //开启分页
                text: {
```

```
                none: '现在这个表格暂无相关数据'
            },
            data: [],
            cols: [
                [ //表头
                    {
                        field: 'id',
                        title: 'ID',
                        width: 200,
                        sort: true,
                        fixed: 'left'
                    }, {
                        field: 'username',
                        title: '账号',
                        width: 200
                    }, {
                        field: 'password',
                        title: '账号',
                        width: 200,
                        sort: true
                    }
                ]
            ]
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。

1.23.3.15 autoSort

autoSort 值类型 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。

1.23.3.15.1 开启与禁用排序

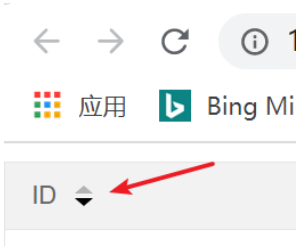
禁用排序的测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                autoSort: false,
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            width: 200,
                            sort: true,
                            fixed: 'left'
                        }, {
                            field: 'username',
                            title: '账号',
                            width: 200
                        }, {
                            field: 'password',
                            title: '账号',
                            width: 200,
                            sort: true
                        }
                    ]
                ]
            });
        </script>
    </body>
</html>
```

启用排序的测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                autoSort: true,
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            width: 200,
                            sort: true,
                            fixed: 'left'
                        }, {
                            field: 'username',
                            title: '账号',
                            width: 200
                        }, {
                            field: 'password',
                            title: '账号',
                            width: 200,
                            sort: true
                        }
                    ]
                ]
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.23.3.15.2 使用服务器端排序

文件 asc.html 代码如下：

```
{
    "code": 0,
    "msg": "",
    "count": 1000,
    "data": [{
        "id": 1,
        "username": "username1",
        "password": "password1"
    },
    {
        "id": 2,
        "username": "username2",
        "password": "password2"
    },
    {
        "id": 3,
        "username": "username3",
        "password": "password3"
    }
    ]
}
```

文件 desc.html 代码如下：

```
{
    "code": 0,
    "msg": "",
    "count": 1000,
    "data": [{
        "id": 3,
        "username": "username3",
        "password": "password3"
    },
```

```
    {
        "id": 2,
        "username": "username2",
        "password": "password2"
    },
    {
        "id": 1,
        "username": "username1",
        "password": "password1"
    }
    ]
}
```

模拟服务端排序的测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'asc.html', //数据接口
                page: true, //开启分页
                autoSort: false, //禁用客户端排序，使用服务端排序
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            width: 200,
                            sort: true,
                            fixed: 'left'
                        }, {
                            field: 'username',
                            title: '账号',
```

```
                    width: 200
                }, {
                    field: 'password',
                    title: '账号',
                    width: 200,
                    sort: true
                }
            ]
        ]
    });

    table.on('sort(test)', function(obj) {
        //sort是工具条事件名，test是table原始容器的属性lay-filter="
对应的值"
        console.log(obj.field); //当前排序的字段名
        console.log(obj.type); //当前排序类型：desc（降序）、asc（升
序）、null（空对象，默认排序）
        console.log(this); //当前排序的th对象

        //尽管table自带排序功能，但并没有请求服务端。
        //有些时候，可能需要根据当前排序的字段，重新向服务端发送请求，
从而实现服务端排序，
        //示例代码如下：

        var sortType = "";
        if (obj.type == 'asc') {
            sortType = "asc.html";
        } else if (obj.type == 'desc') {
            sortType = "desc.html";
        } else if (obj.type == null) {
            sortType = "asc.html";
        }

        table.reload('demo', {
            initSort: obj, //记录初始排序，如果不设的话，将无法标记表
头的排序状态。
            url: sortType
            //where: { //请求参数（注意：这里面的参数可任意定义，并非
下面固定的格式）
            /// field: obj.field, //排序字段
            //  order: obj.type //排序方式
            //}
        });
        layer.msg('服务端排序，order by ' + obj.field + ' ' + obj.type);
```

```
            });
        </script>
    </body>
</html>
```

注意：点击排序列时有 3 个值：asc 正排序，desc 倒排序，null 默认值。

1.23.3.16 initSort

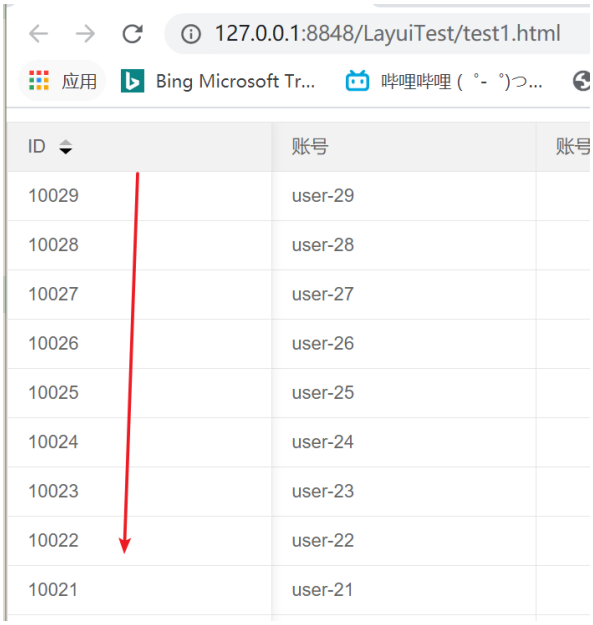
initSort 值类型 Object。初始排序状态，用于在数据表格渲染完毕时，默认按某个字段排序。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                autoSort: true,
                initSort: {
                    field: 'id', //排序字段，对应 cols 设定的各字段名
                    type: 'desc' //排序方式  asc: 升序、desc: 降序、null: 默认排序
                },
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            width: 200,
                            sort: true,
                            fixed: 'left'
```

```
                    }, {
                        field: 'username',
                        title: '账号',
                        width: 200
                    }, {
                        field: 'password',
                        title: '账号',
                        width: 200,
                        sort: true
                    }
                ]
            ]
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.23.3.17 id

id 值类型 String。设定容器唯一 id。id 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。该参数也可以自动从<table id="test"></table>中的 id 参数中获取。

1.23.3.18 设定表格外观

控制表格外观的主要参数由如图 1-xx 所示组成：

参数名	可选值	备注
skin	line （行边框风格） row （列边框风格） nob （无边框风格）	用于设定表格风格，若使用默认风格不设置该属性即可
even	true/false	若不开启隔行背景，不设置该参数即可
size	sm （小尺寸） lg （大尺寸）	用于设定表格尺寸，若使用默认尺寸不设置该属性即可

示例代码如下：

```
//“方法级渲染”配置方式
table.render({ //其它参数在此省略
  skin: 'line', //行边框风格
  even: true, //开启隔行背景
  size: 'sm' //小尺寸的表格
});
```

等价于（“自动化渲染”配置方式）

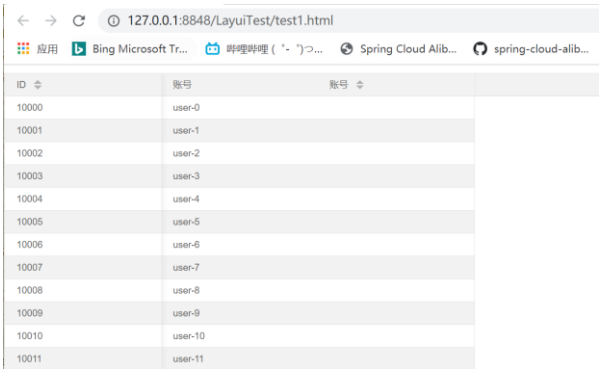
<table class="layui-table" lay-data="{skin:'line', even:true, size:'sm'}" lay-filter="test"> …… </table>

示例代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                skin: 'line', //行边框风格
                even: true, //开启隔行背景
                size: 'sm', //小尺寸的表格
```

```
            cols: [
                [ //表头
                    {
                        field: 'id',
                        title: 'ID',
                        width: 200,
                        sort: true,
                        fixed: 'left'
                    }, {
                        field: 'username',
                        title: '账号',
                        width: 200
                    }, {
                        field: 'password',
                        title: '账号',
                        width: 200,
                        sort: true
                    }
                ]
            ]
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.23.4 异步数据接口

数据的异步请求由图 1-xx 所示的参数组成。

参数名	功能
url	接口地址。默认会自动传递两个参数：*?page=1&limit=30*（该参数可通过 request 自定义） *page* 代表当前页码、 *limit* 代表每页数据量
method	接口http请求类型，默认：get
where	接口的其它参数。如： *where: {token: 'sasasas', id: 123}*
contentType	发送到服务端的内容编码类型。如果你要发送 json 内容，可以设置： *contentType: 'application/json'*
headers	接口的请求头。如： *headers: {token: 'sasasas'}*

parseData

数据格式解析的回调函数，用于将返回的任意数据格式解析成 table 组件规定的数据格式。

table 组件默认规定的数据格式为（参考：/demo/table/user/）：

默认规定的数据格式 layui.code

```
{
  "code": 0,
  "msg": "",
  "count": 1000,
  "data": [{}, {}]
}
```

很多时候，您接口返回的数据格式并不一定都符合 table 默认规定的格式，比如：

假设您返回的数据格式 layui.code

```
{
  "status": 0,
  "message": "",
  "total": 180,
  "data": {
    "item": [{}, {}]
  }
}
```

那么你需要借助 parseData 回调函数将其解析成 table 组件所规定的数据格式

code layui.code

```
table.render({
  elem: '#demp'
  ,url: ''
  ,parseData: function(res){ //res 即为原始返回的数据
    return {
      "code": res.status, //解析接口状态
      "msg": res.message, //解析提示文本
      "count": res.total, //解析数据长度
      "data": res.data.item //解析数据列表
    };
  }
  //,…… //其他参数
});
```

该参数非常实用，系 layui 2.4.0 开始新增

request

用于对分页请求的参数：page、limit重新设定名称，如：

```
table.render({
  elem: '#demp'
  ,url: ''
  ,request: {
    pageName: 'curr' //页码的参数名称，默认：page
    ,limitName: 'nums' //每页数据量的参数名，默认：limit
  }
  //,…… //其他参数
});
```

那么请求数据时的参数将会变为： *?curr=1&nums=30*

response

您还可以借助 response 参数来重新设定返回的数据格式，如：

```
table.render({
  elem: '#demp'
  ,url: ''
  ,response: {
    statusName: 'status' //规定数据状态的字段名称，默认：code
    ,statusCode: 200 //规定成功的状态码，默认：0
    ,msgName: 'hint' //规定状态信息的字段名称，默认：msg
    ,countName: 'total' //规定数据总数的字段名称，默认：count
    ,dataName: 'rows' //规定数据列表的字段名称，默认：data
  }
  //,…… //其他参数
});
```

那么上面所规定的格式为：

重新规定的数据格式

```
{
  "status": 200,
  "hint": "",
  "total": 1000,
  "rows": []
}
```

使用示例代码如下：

```
//“方法级渲染”配置方式
table.render({ //其它参数在此省略
   url: '/api/data/'
   //where: {token: 'sasasas', id: 123} //如果无需传递额外参数，可不加该参数
   //method: 'post' //如果无需自定义 HTTP 类型，可不加该参数
   //request: {} //如果无需自定义请求参数，可不加该参数
   //response: {} //如果无需自定义数据响应名称，可不加该参数
});

等价于（“自动化渲染”配置方式）
<table class="layui-table" lay-data="{url:'/api/data/'}" lay-filter="test"> …… </table>
```

1.23.4.1 发起 get 请求到 Servlet

Servlet 代码如下：

```
package controller;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/Test1")
public class Test1 extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        System.out.println("Test1 run doGet !");
    }
}
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'Test1',
                method: "get",
                page: true,
                cols: [
                    [ //表头
```

```
                    {
                        field: 'id',
                        title: 'ID',
                        width: 200,
                        sort: true,
                        fixed: 'left'
                    }, {
                        field: 'username',
                        title: '账号',
                        width: 200
                    }, {
                        field: 'password',
                        title: '账号',
                        width: 200,
                        sort: true
                    }
                ]
            ]
        });
    </script>
</body>
</html>
```

在 F12 中显示的运行结果如图 1-xx 所示。

1.23.4.2 发起 post 请求到 Servlet

Servlet 代码如下：

```
package controller;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/Test2")
public class Test2 extends HttpServlet {
```

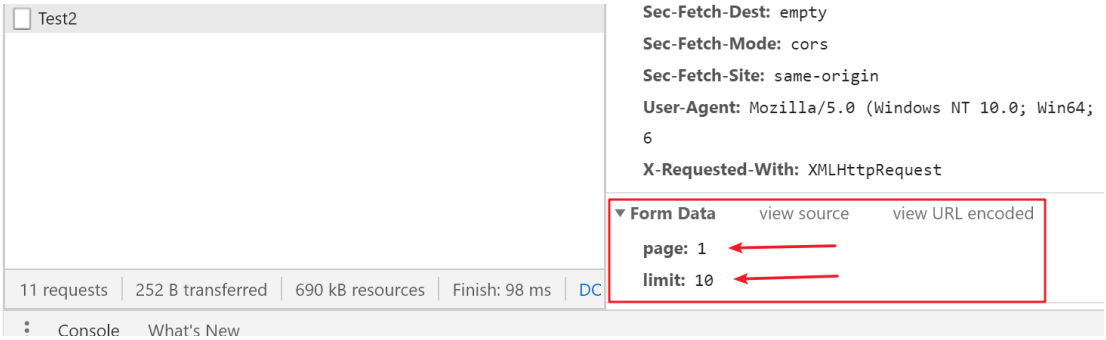
```
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        System.out.println("Test2 run doPost !");
    }
}
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'Test2',
                method: "post",
                page: true,
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            width: 200,
                            sort: true,
                            fixed: 'left'
                        }, {
                            field: 'username',
                            title: '账号',
                            width: 200
                        }, {
                            field: 'password',
                            title: '账号',
                            width: 200,
                            sort: true
```

```
                    }
                ]
            ]
        });
    </script>
</body>
</html>
```

在 F12 中显示的运行结果如图 1-xx 所示。



1.23.4.3 使用 where 属性

Servlet 代码如下：

```
package controller;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/Test3")
public class Test3 extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        System.out.println("Test3 run doPost !");

        String username = request.getParameter("username");
        String password = request.getParameter("password");

        System.out.println(username + " " + password);
    }
```

```
}
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'Test3',
                method: "post",
                where: {
                    "username": "中国",
                    "password": "中国人"
                },
                page: true,
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            width: 200,
                            sort: true,
                            fixed: 'left'
                        }, {
                            field: 'username',
                            title: '账号',
                            width: 200
                        }, {
                            field: 'password',
                            title: '账号',
                            width: 200,
                            sort: true
                        }
```

```
                ]
            ]
        });
    </script>
</body>
</html>
```

1.23.4.4 使用 contentType: 'application/json'传递 JSON

Servlet 代码如下：

```
package controller;

import java.io.IOException;
import java.io.InputStream;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/Test4")
public class Test4 extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        request.setCharacterEncoding("utf-8");
        System.out.println("Test4 run doPost !");
        int totalBytes = request.getContentLength();
        InputStream inputStream = request.getInputStream();
        byte[] byteArray = new byte[totalBytes];
        inputStream.read(byteArray);
        System.out.println(new String(byteArray, "utf-8"));
    }
}
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
```

```
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'Test4',
                method: "post",
                where: {
                    "username": "中国",
                    "password": "中国人"
                },
                contentType: 'application/json;',
                page: true,
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            width: 200,
                            sort: true,
                            fixed: 'left'
                        }, {
                            field: 'username',
                            title: '账号',
                            width: 200
                        }, {
                            field: 'password',
                            title: '账号',
                            width: 200,
                            sort: true
                        }
                    ]
                ]
            });
        </script>
    </body>
</html>
```

1.23.4.5 使用 headers 属性

Servlet 代码如下：

```
package controller;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/Test5")
public class Test5 extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        request.setCharacterEncoding("utf-8");
        System.out.println("Test5 run doPost !");
        System.out.println(request.getHeader("token"));
        System.out.println(request.getHeader("mykey"));
    }
}
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'Test5',
                headers: {
```

```
                "token": "123456789",
                "mykey": "myvalue"
            },
            method: "post",
            page: true,
            cols: [
                [ //表头
                    {
                        field: 'id',
                        title: 'ID',
                        width: 200,
                        sort: true,
                        fixed: 'left'
                    }, {
                        field: 'username',
                        title: '账号',
                        width: 200
                    }, {
                        field: 'password',
                        title: '账号',
                        width: 200,
                        sort: true
                    }
                ]
            ]
        });
    </script>
</body>
</html>
```

1.23.4.6 table 处理服务端响应的数据

本实验依赖如下 jar 文件：
jackson-annotations-2.11.0.jar
jackson-core-2.11.0.jar
jackson-databind-2.11.0.jar

实体类 Userinfo.java 代码如下：

```
package entity;

public class Userinfo {
    private int id;
    private String username;
    private String password;
```

```
    public Userinfo() {
    }

    public Userinfo(int id, String username, String password) {
        super();
        this.id = id;
        this.username = username;
        this.password = password;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

}
```

DTO 类代码如下：

```
package dto;

import java.util.List;

public class LayuiTableBox {
    private int code;
```

```
    private String msg;
    private int count;
    private List data;

    public LayuiTableBox() {
    }

    public LayuiTableBox(int code, String msg, int count, List data) {
        super();
        this.code = code;
        this.msg = msg;
        this.count = count;
        this.data = data;
    }

    public int getCode() {
        return code;
    }

    public void setCode(int code) {
        this.code = code;
    }

    public String getMsg() {
        return msg;
    }

    public void setMsg(String msg) {
        this.msg = msg;
    }

    public int getCount() {
        return count;
    }

    public void setCount(int count) {
        this.count = count;
    }

    public List getData() {
        return data;
    }

    public void setData(List data) {
```

```
        this.data = data;
    }

}
```

Table 组件需要如下 4 个固定的属性名，建议不要更改。

```
    private int code;
    private String msg;
    private int count;
    private List data;
```

Servlet 类代码如下：

```
package controller;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.fasterxml.jackson.databind.ObjectMapper;

import dto.LayuiTableBox;
import entity.Userinfo;

@WebServlet("/Test6")
public class Test6 extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        request.setCharacterEncoding("utf-8");
        System.out.println("Test6 run doPost !");

        String username = request.getParameter("username");
        String password = request.getParameter("password");
        System.out.println(username);
        System.out.println(password);

        List list = new ArrayList();
```

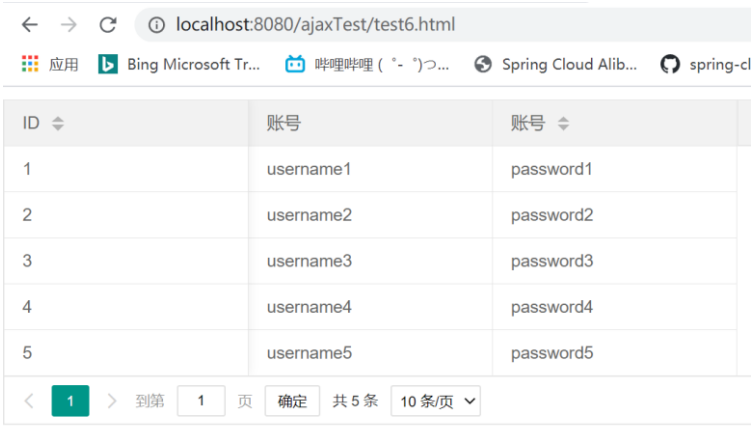
```
        list.add(new Userinfo(1, "username1", "password1"));
        list.add(new Userinfo(2, "username2", "password2"));
        list.add(new Userinfo(3, "username3", "password3"));
        list.add(new Userinfo(4, "username4", "password4"));
        list.add(new Userinfo(5, "username5", "password5"));

        LayuiTableBox box = new LayuiTableBox();
        box.setCode(0);
        box.setCount(5);
        box.setData(list);

        ObjectMapper mapper = new ObjectMapper();
        String jsonString = mapper.writeValueAsString(box);

        System.out.println(jsonString);

        response.setContentType("application/json;charset=utf-8");
        PrintWriter out = response.getWriter();
        out.print(jsonString);
        out.flush();
        out.close();

    }
}
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'Test6',
                where: {
```

```
                "username": "中国",
                "password": "中国人"
            },
            method: "post",
            page: true,
            cols: [
                [ //表头
                    {
                        field: 'id',
                        title: 'ID',
                        width: 200,
                        sort: true,
                        fixed: 'left'
                    }, {
                        field: 'username',
                        title: '账号',
                        width: 200
                    }, {
                        field: 'password',
                        title: '账号',
                        width: 200,
                        sort: true
                    }
                ]
            ]
        });
    </script>
</body>
</html>
```

运行结果如图 1-xx 所示。



1.23.5 cols 表头参数一览表

cols 表头参数如图 1-xx 所示。

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/Object	是否开启该列的自动合计功能，默认：false。 当开启时，则默认由前端自动合计当前行数据。从 layui 2.5.6 开始：若接口直接返回了合计行数据，则优先读取接口合计行数据，格式如下： code layui.code `{ "code": 0, "msg": "", "count": 1000, "data": [{}, {}] "totalRow": { "score": "666" ,"experience": "999" } }` 如上，在 totalRow 中返回所需统计的列字段名和值即可。 另外，totalRow 字段同样可以通过 parseData 回调来解析成为 table 组件所规定的数据格式。	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 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 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
toolbar	String	绑定工具条模板。可在每行对应的列中出现一些自定义的操作性按钮	详见行工具事件

1.23.5.1 width

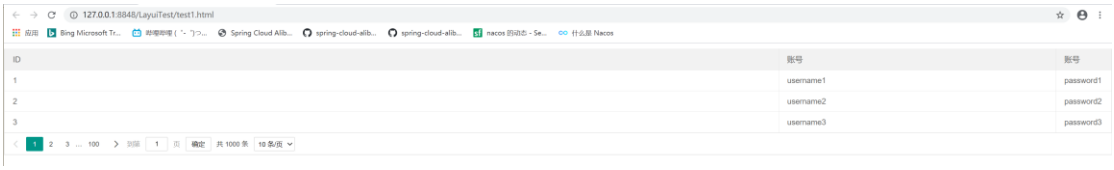
width 值类型 Number/String。设定列宽，若不填写则自动分配；若填写，则支持值为：数字、百分比

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
```

```
                    field: 'id',
                    title: 'ID',
                }, {
                    field: 'username',
                    title: '账号',
                    width: 500
                }, {
                    field: 'password',
                    title: '账号',
                    width: "5%"
                }
            ]
        ]
    });
</script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.23.5.2 minWidth

minWidth 值类型 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。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
```

```
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                        }, {
                            field: 'username',
                            title: '账号',
                            minWidth: 500
                        }, {
                            field: 'password',
                            title: '账号'
                        }
                    ]
                ]
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.23.5.3 type

type 值类型 String，设定列类型，可选值有：

（1）normal（常规列，无需设定）
（2）checkbox（复选框列）
（3）radio（单选框列）
（4）numbers（序号列）
（5）space（空列）

1.23.5.3.1 checkbox

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
                            field: '',
                            title: '',
                            type: "checkbox"
                        },
                        {
                            field: 'id',
                            title: 'ID',
                        }, {
                            field: 'username',
                            title: '账号',
                            minWidth: 500
                        }, {
                            field: 'password',
                            title: '账号'
                        }
                    ]
                ]
            });
        </script>
    </body>
```

```
</html>
```

运行效果如图 1-xx 所示。

1.23.5.3.2 radio

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
                            field: '',
                            title: '',
                            type: "radio"
                        },
                        {
                            field: 'id',
                            title: 'ID',
                        }, {
```

```
                    field: 'username',
                    title: '账号',
                    minWidth: 500
                }, {
                    field: 'password',
                    title: '账号'
                }
            ]
        ]
    });
</script>
</body>
</html>
```

运行效果如图 1-xx 所示。

1.23.5.3.3 numbers

测试代码如下：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title></title>
    <script src="js/jquery3.5.1.js"></script>
    <link rel="stylesheet" href="layui/css/layui.css" />
    <script src="layui/layui.all.js"></script>
</head>
<body>
    <table id="demo" lay-filter="test"></table>
    <script>
        var table = layui.table;
        //第一个实例
        table.render({
            elem: '#demo',
            url: 'new.html', //数据接口
```

```
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
                            field: '',
                            title: '',
                            type: "numbers"
                        },
                        {
                            field: 'id',
                            title: 'ID',
                        }, {
                            field: 'username',
                            title: '账号',
                            minWidth: 500
                        }, {
                            field: 'password',
                            title: '账号'
                        }
                    ]
                ]
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.23.5.3.4 space

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
```

```
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
                            field: '',
                            title: '',
                            type: "space",
                            width: 100
                        },
                        {
                            field: 'id',
                            title: 'ID',
                        }, {
                            field: 'username',
                            title: '账号',
                            minWidth: 500
                        }, {
                            field: 'password',
                            title: '账号'
                        }
                    ]
                ]
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.23.5.4 LAY_CHECKED

LAY_CHECKED 值类型 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。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
                            field: '',
                            title: '',
                            type: "checkbox",
                            LAY_CHECKED: true
                        },
                        {
                            field: 'id',
```

```
                    title: 'ID',
                }, {
                    field: 'username',
                    title: '账号',
                    minWidth: 500
                }, {
                    field: 'password',
                    title: '账号'
                }
            ]
        ]
    });
</script>
</body>
</html>
```

运行效果如图 1-xx 所示。

1.23.5.5 fixed

fixed 值类型 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。

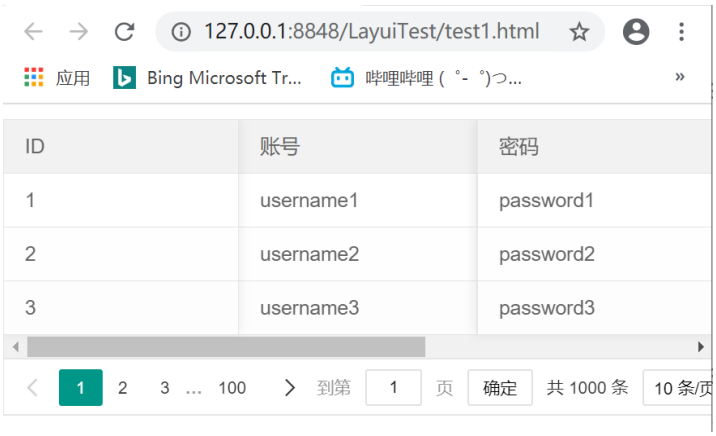
注意：如果是固定在左，该列必须放在表头最前面；如果是固定在右，该列必须放在表头最后面。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
```

```
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            fixed: "left"
                        }, {
                            field: 'username',
                            title: '账号',
                            minWidth: 500
                        }, {
                            field: 'password',
                            title: '密码',
                            fixed: "right"
                        }
                    ]
                ]
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.23.5.6 hide

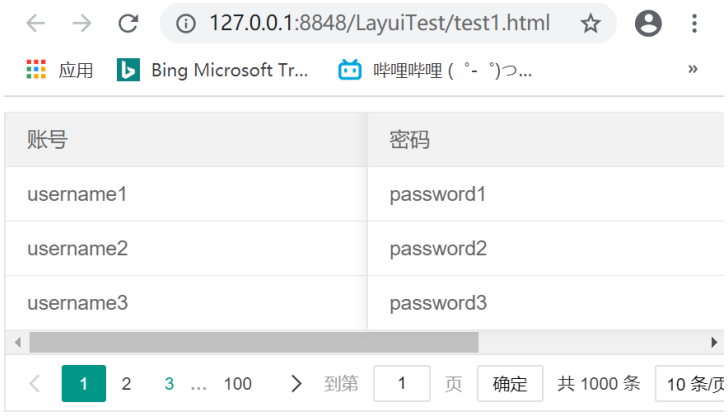
hide 值类型 Boolean。是否初始隐藏列，默认：false。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            fixed: "left",
                            hide: true
                        }, {
                            field: 'username',
                            title: '账号',
                            minWidth: 500
                        }, {
                            field: 'password',
                            title: '密码',
                            fixed: "right"
                        }
                    ]
                ]
            });
        </script>
    </body>
```

```
</html>
```

运行效果如图 1-xx 所示。



1.23.5.7 totalRow

totalRow 值类型 Boolean/Object。是否开启该列的自动合计功能，默认：false。

当开启时，则默认由前端自动合计当前行数据。若接口直接返回了合计行数据，则优先读取接口合计行数据，格式如下：

```
{
  "code": 0,
  "msg": "",
  "count": 1000,
  "data": [{}, {}]
  "totalRow": {
    "score": "666"
    ,"experience": "999"
  }
}
```

在 totalRow 中返回所需统计的列字段名和值即可。

另外，totalRow 字段同样可以通过 parseData 回调来解析成为 table 组件所规定的数据格式。

文件 new.html 代码如下：

```
{
    "code": 0,
    "msg": "",
    "count": 1000,
    "data": [{
            "id": 1,
            "username": "username1",
            "password": "password1"
        },
```

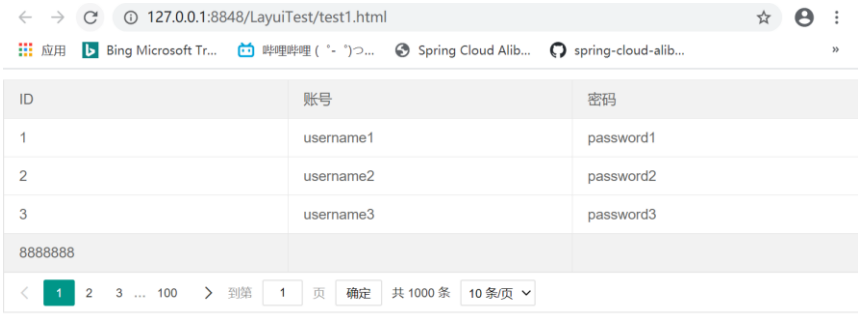
```
        {
            "id": 2,
            "username": "username2",
            "password": "password2"
        },
        {
            "id": 3,
            "username": "username3",
            "password": "password3"
        }
    ],
    "totalRow": {
        "id": "8888888"
    }
}
```

运行测试文件代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                totalRow: true,
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                        }, {
                            field: 'username',
                            title: '账号',
```

```
                    }, {
                        field: 'password',
                        title: '密码',
                    }
                ]
            ]
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.23.5.8 totalRowText

totalRowText 值类型 String。用于显示自定义的合计文本。

文件 new.html 代码如下：

```
{
    "code": 0,
    "msg": "",
    "count": 1000,
    "data": [{
            "id": 1,
            "username": "username1",
            "password": "password1"
        },
        {
            "id": 2,
            "username": "username2",
            "password": "password2"
        },
        {
            "id": 3,
            "username": "username3",
```

```
            "password": "password3"
        }
    ]
}
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                totalRow: true,
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            totalRowText: "999999999"
                        }, {
                            field: 'username',
                            title: '账号',
                        }, {
                            field: 'password',
                            title: '密码',
                        }
                    ]
                ]
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.23.5.9 sort

sort 值类型 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'，排序后可能并不是想要的结果。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                totalRow: true,
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
```

```
                    sort: true
                }, {
                    field: 'username',
                    title: '账号',
                }, {
                    field: 'password',
                    title: '密码',
                }
            ]
        ]
    });
</script>
</body>
</html>
```

运行效果如图 1-xx 所示。

1.23.5.10 unresize

unresize 值类型 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，当然也可以设置 true 来禁用该功能。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
```

```
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                totalRow: true,
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                        }, {
                            field: 'username',
                            title: '账号',
                            unresize: true
                        }, {
                            field: 'password',
                            title: '密码',
                        }
                    ]
                ]
            });
        </script>
    </body>
</html>
```

1.23.5.11 edit

edit 值类型 String。单元格编辑类型（默认不开启）目前只支持：text（输入框）。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
```

```
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                totalRow: true,
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                        }, {
                            field: 'username',
                            title: '账号',
                            edit: "text"
                        }, {
                            field: 'password',
                            title: '密码',
                        }
                    ]
                ]
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.23.5.12 event

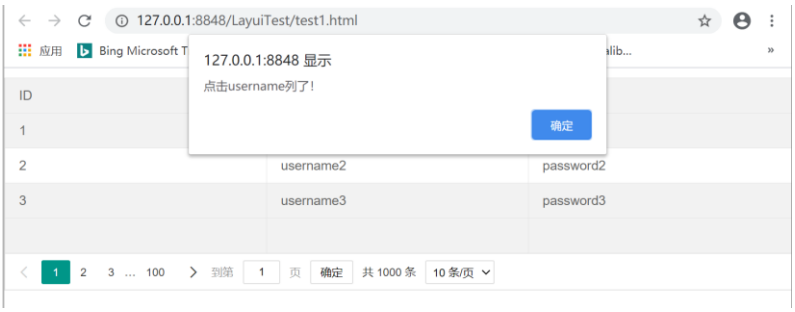
event 值类型 String。自定义单元格点击事件名，以便在 tool 事件中完成对该单元格的业务处理。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                totalRow: true,
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                        }, {
                            field: 'username',
                            title: '账号',
                            event: "clickUsername",

                        }, {
                            field: 'password',
                            title: '密码',
                        }
                    ]
                ]
            });


            //监听工具条
            table.on('tool(test)', function(obj) { //注：tool是工具条事件名，
test是table原始容器的属性lay-filter="对应的值"
                var data = obj.data; //获得当前行数据
```

```
                var layEvent = obj.event; //获得lay-event对应的值（也可以是表头的event参数对应的值）
                var tr = obj.tr; //获得当前行tr的DOM对象（如果有的话）

                if (layEvent === 'clickUsername') { //查看
                    alert('点击username列了！ ');
                }
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.23.5.13 style

style 值类型 String。自定义单元格样式。即传入 CSS 样式

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
```

```
                page: true, //开启分页
                totalRow: true,
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                        }, {
                            field: 'username',
                            title: '账号',
                            style: "color:red;font-size:30px",
                        }, {
                            field: 'password',
                            title: '密码',
                        }
                    ]
                ]
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.23.5.14 align

align 值类型 String。单元格排列方式，可选值有：left（默认）、center（居中）、right（居右）。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
```

```
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                totalRow: true,
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            align: "left",
                        }, {
                            field: 'username',
                            title: '账号',
                            align: "center",
                        }, {
                            field: 'password',
                            title: '密码',
                            align: "right",
                        }
                    ]
                ]
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.23.5.15 colspan 和 rowspan

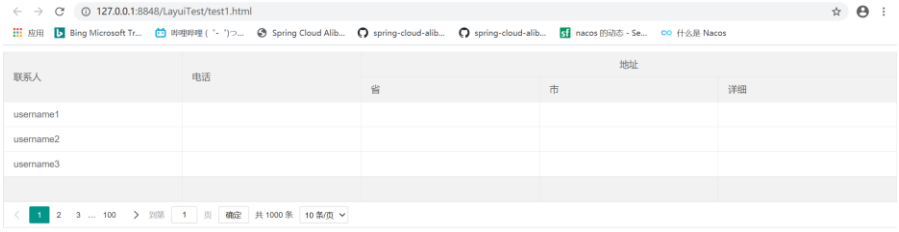
colspan 值类型 Number。单元格所占列数（默认：1）。一般用于多级表头
rowspan 值类型 Number。单元格所占行数（默认：1）。一般用于多级表头

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                totalRow: true,
                cols: [
                    [ //标题栏
                        {
                            field: 'username',
                            title: '联系人',
                            rowspan: 2
                        } //rowspan即纵向跨越的单元格数
                        , {
                            field: 'phone',
                            title: '电话',
                            rowspan: 2
                        }, {
                            align: 'center',
                            title: '地址',
                            colspan: 3
                        } //colspan即横跨的单元格数，这种情况下不用设置field和
width
                    ],
                    [{
```

```
                    field: 'province',
                    title: '省',
                }, {
                    field: 'city',
                    title: '市',
                }, {
                    field: 'county',
                    title: '详细',
                }]
            ]
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.23.5.16 templet

templet 值类型 String，默认值无。自定义列模板，模板遵循 laytpl 语法。在默认情况下，单元格的内容是完全按照数据接口返回的 content 原样输出的，如果想对某列的单元格添加链接等其它元素，可以借助该参数来轻松实现，这是一个非常实用且强大的功能，表格内容会因此而丰富多样。

templet 提供了三种使用方式，请结合实际场景选择最合适的一种：

（1）如果自定义模版的字符量太大，推荐采用【方式一】；

（2）如果自定义模板的字符量适中，或者想更方便地调用外部方法，推荐采用【方式二】；

（3）如果自定义模板的字符量很小，推荐采用【方式三】；

1.23.5.16.1 方式一：绑定模版选择器

示例代码如下：

```
table.render({
  cols: [[
    {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
    ,{field:'id', title:'ID', width:100}
  ]]
```

```
});
```

等价于：

```
<th lay-data="{field:'title', width: 200, templet: '#titleTpl'}">文章标题</th>
<th lay-data="{field:'id', width:100}">ID</th>
```

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 语法，可读取到返回的所有数据，示例代码如下：

```
<script type="text/html" id="titleTpl">
  <a href="/detail/{{d.id}}" class="layui-table-link">{{d.title}}</a>
</script>
```

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，还可以读取到以下额外字段：序号{{d.LAY_INDEX}}。

由于模板遵循 laytpl 语法（建议细读 laytpl 文档），因此在模板中可以写任意脚本语句（如 if else/for 等），示例代码如下：

```
<script type="text/html" id="titleTpl">
  {{#  if(d.id < 100){ }}
    <a href="/detail/{{d.id}}" class="layui-table-link">{{d.title}}</a>
  {{#  } else { }}
    {{d.title}}(普通用户)
  {{#  } }}
</script>
```

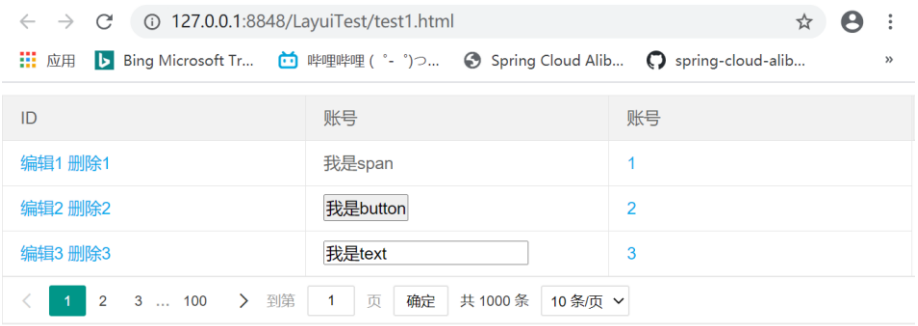
测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>

        <script>
            function editPassword(userId){
              alert(userId);
            }
        </script>

        <script type="text/html" id="titleTpl_1">
```

```
            <a href="/editUserinfo/{{d.id}}" class="layui-table-link">编辑
{{d.id}}</a>
            <a href="/deleteUserinfo/{{d.id}}" class="layui-table-link">
删除{{d.id}}</a>
        </script>

        <script type="text/html" id="titleTpl_2">
            {{# if (d.username=='username1'){ }}
            <span style="red:red">我是span</span>
            {{# } }}
            {{# if (d.username=='username2'){ }}
            <input type="button" value="我是button"/>
            {{# } }}
            {{# if (d.username=='username3'){ }}
            <input type="text" value="我是text"/>
            {{# } }}
        </script>

        <script type="text/html" id="titleTpl_3">
            <a href="javascript:editPassword({{d.id}})"
class="layui-table-link">{{d.id}}</a>
        </script>

        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            templet: '#titleTpl_1'
                        }, {
                            field: 'username',
                            title: '账号',
                            templet: '#titleTpl_2'
                        }, {
                            field: 'password',
                            title: '账号',
                            templet: '#titleTpl_3'
```

```
                    }
                ]
            ]
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.23.5.16.2 方式二：函数转义

templet 支持函数形式，函数返回一个参数 d，包含接口返回的所有字段和数据，示例代码如下：

```
table.render({
  cols: [[
    {field:'title', title: '文章标题', width: 200
      ,templet: function(d){
        return 'ID：'+ d.id +'， 标题：<span style="color: #c00;">'+ d.title +'</span>'
      }
    }
    ,{field:'id', title:'ID', width:100}
  ]]
});
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
```

```
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>

        <script>
            function editPassword(userId) {
                alert(userId);
            }
        </script>

        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            width: 200 //需要加此属性，不然出现bug，显示出下拉
框。
                        }, {
                            field: 'username',
                            title: '账号',
                        }, {
                            field: 'password',
                            title: '账号',
                            templet: function(d) {
                                return "<input type='text' value='" +
d.password + "'/>";
                            }
                        }
                    ]
                ]
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.23.5.16.3 方式三：直接赋值模版字符

事实上，templet 也可以直接是一段 html 内容，示例代码如下：

templet: ' {{d.title}} '

注意：这里一定要被一层 包裹，否则无法读取到模板。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>

        <script>
            function editPassword(userId) {
                alert(userId);
            }
        </script>

        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
```

```
                        field: 'id',
                        title: 'ID',
                        width: 200
                    }, {
                        field: 'username',
                        title: '账号',
                    }, {
                        field: 'password',
                        title: '账号',
                        templet: "<div><span
style='color:blue'>{{d.password}}</span></div>"
                    }
                ]
            ]
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。

1.23.5.17 toolbar

类型：String，默认值无。toolbar 值类型 String 绑定工具条模板，可以在每行对应的列中出现一些自定义的操作性按钮。

如果需要在表格的每一行加上“查看”、“编辑”、“删除”这样类似的操作按钮，tool 参数就是为此而生，因此可以非常便捷地实现各种操作功能。tool 参数和 templet 参数的使用方式完全类似，通常接受的是一个选择器，也可以是一段 HTML 字符，示例代码如下：

```
table.render({
  cols: [[
    {field:'id', title:'ID', width:100}
    ,{fixed: 'right', width:150, align:'center', toolbar: '#barDemo'} //这里的 toolbar 值是模板元素的选择器
```

```
    ]]
});
```

等价于：

```
<th lay-data="{field:'id', width:100}">ID</th>
<th lay-data="{fixed: 'right', width:150, align:'center', toolbar: '#barDemo'}"></th>
```

下述是 toolbar 对应的模板，它可以存放在页面的任意位置：

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

注意：lay-event=""属性值是模板的关键所在，值可随意定义。

接下来可以借助 table 模块的工具条事件完成不同的操作功能：

```
//监听工具条
table.on('tool(test)', function(obj) {
    //注：tool是工具条事件名，test是table原始容器的属性lay-filter="对应的值"
    var data = obj.data; //获得当前行数据
    var layEvent = obj.event; //获得lay-event对应的值（也可以是表头的event参数对应的值）
    var tr = obj.tr; //获得当前行tr的DOM对象（如果有的话）

    if (layEvent === 'detail') { //查看
        //do somehing
    } else if (layEvent === 'del') { //删除
        layer.confirm('真的删除行么', function(index) {
            obj.del(); //删除对应行（tr）的DOM结构，并更新缓存
            layer.close(index);
            //向服务端发送删除指令
        });
    } else if (layEvent === 'edit') { //编辑
        //do something

        //同步更新缓存对应的值
        obj.update({
            username: '123',
            title: 'xxx'
```

```
        });
    }
});
```

右上角自定义按钮配置代码如下：

```
table.render({ //其它参数在此省略
    defaultToolbar: ['filter', 'print', 'exports', {
        title: '提示', //标题
        layEvent: 'LAYTABLE_TIPS', //事件名，用于toolbar事件中使用
        icon: 'layui-icon-tips' //图标类名
    }]
});
```

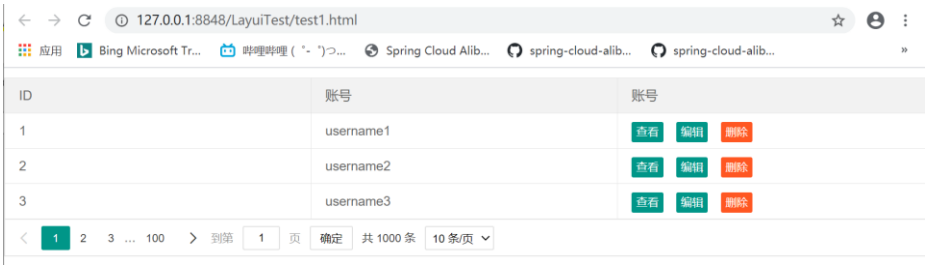
测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>

        <script type="text/html" id="barDemo">
            <a class="layui-btn layui-btn-xs" lay-event="detail">查看</a>
            <a class="layui-btn layui-btn-xs" lay-event="edit">编辑</a>
            <a class="layui-btn layui-btn-danger layui-btn-xs"
lay-event="del">删除</a>
        </script>

        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                toolbar: true,
                defaultToolbar: ['filter', 'print', 'exports', {
                    title: '我是自定义按钮', //标题
                    layEvent: 'myevent', //事件名，用于toolbar事件中使用
```

```
                icon: 'layui-icon-tips' //图标类名
            }],
            cols: [
                [ //表头
                    {
                        field: 'id',
                        title: 'ID',
                        width: 200
                    }, {
                        field: 'username',
                        title: '账号',
                    }, {
                        field: 'password',
                        title: '账号',
                        toolbar: '#barDemo'
                    }
                ]
            ]
        });

        //监听工具条
        table.on('toolbar(test)', function(obj) {
            var layEvent = obj.event;
            if (layEvent === 'myevent') {
                alert("自定义事件myevent");
            }
        });

        //监听工具条
        table.on('tool(test)', function(obj) { //注：tool 是工具条事件名，test 是 table 原始容器的属性 lay-filter="对应的值"
            var data = obj.data; //获得当前行数据
            var layEvent = obj.event; //获得 lay-event 对应的值（也可以是表头的 event 参数对应的值）
            var tr = obj.tr; //获得当前行 tr 的 DOM 对象（如果有的话）

            if (layEvent === 'detail') { //查看
                alert("查看" + data.id);
            } else if (layEvent === 'del') { //删除
                layer.confirm('真的删除记录' + data.id + "吗？", function(index) {
                    obj.del(); //删除对应行（tr）的DOM结构，并更新缓存
                    layer.close(index);
                    //向服务端发送删除指令
```

```
                });
            } else if (layEvent === 'edit') { //编辑
                alert("编辑" + data.id);
                //同步更新缓存对应的值
                obj.update({
                    username: 'xxxxxxxxxxx'
                });
            }
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.23.6 基础方法

基础用法是 table 模块的关键组成部分，目前所开放的所有方法如图 1-xx 所示。

```
code
> table.set(options); //设定全局默认参数。options即各项基础参数
> table.on('event(filter)', callback); //事件监听。event为内置事件名（详见下文），filter为容器lay-filter设定的值
> table.init(filter, options); //filter为容器lay-filter设定的值，options即各项基础参数。例子见：转换静态表格
> table.checkStatus(id); //获取表格选中行（下文会有详细介绍）。id 即为 id 参数对应的值
> table.render(options); //用于表格方法级渲染，核心方法。应该不用再过多解释了，详见：方法级渲染
> table.reload(id, options); //表格重载
> table.resize(id); //重置表格尺寸
> table.exportFile(id, data, type); //导出数据
```

后面的章节会对常用的方法进行案例演示。

1.23.7 获取行中 checkbox 状态

可获取到表格所有的选中行相关数据。

语法：table.checkStatus('ID');

其中 ID 为基础参数 id 对应的值，示例代码如下：

【自动化渲染】

<table class="layui-table" lay-data="{id: 'idTest'}"> …… </table>

【方法渲染】

```
table.render({ //其它参数省略
  id: 'idTest'
});
```

javascript 调用代码如下：

```
var checkStatus = table.checkStatus('idTest'); //idTest 即为基础参数 id 对应的值
console.log(checkStatus.data) //获取选中行的数据
console.log(checkStatus.data.length) //获取选中行数量，可作为是否有选中行的条件
console.log(checkStatus.isAll ) //表格是否全选
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>

        <script type="text/html" id="barDemo">
            <a class="layui-btn layui-btn-xs" lay-event="detail">查看</a>
            <a class="layui-btn layui-btn-xs" lay-event="edit">编辑</a>
            <a class="layui-btn layui-btn-danger layui-btn-xs"
lay-event="del">删除</a>
        </script>

        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                id: "mytable",
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
                            type: "checkbox",
```

```
                    width:100
                },
                {
                    field: 'id',
                    title: 'ID',
                    width:200
                }, {
                    field: 'username',
                    title: '账号',
                    width:200
                }, {
                    field: 'password',
                    title: '账号',
                    width:200
                }
            ]
        ]
    });

    function getCheckedStatus(){
        var checkStatus = table.checkStatus('mytable'); //mytable
即为基础参数id对应的值
        console.log(checkStatus.data); //获取选中行的数据
        console.log(checkStatus.data.length); //获取选中行数量，可作
为是否有选中行的条件
        console.log(checkStatus.isAll); //表格是否全选
    }
</script>
<input type="button" onclick="javascript:getCheckedStatus()"
value="取值" />
</body>
</html>
```

运行效果如图 1-xx 所示。

1.23.8 重置表格尺寸

可重置表格尺寸和结构，其内部完成了：固定列高度平铺、动态分配列宽、容器滚动条宽高补丁等操作。它一般用于特殊情况下（如“非窗口 resize”导致的表格父容器宽度变化而引发的列宽适配异常），以保证表格在此类特殊情况下依旧能正常展示。

语法：table.resize('ID')，其中 ID 为基础参数 id 对应的值，示例代码如下：

```
table.render({ //其它参数省略
   ,elem: '#demo'
   //,…… //其它参数
   ,id: 'idTest'
});

//执行表格“尺寸结构”的重置，一般写在对应的事件中
table.resize('idTest');
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div style="background-color: red;">
            <table id="demo" lay-filter="test"></table>
        </div>

        <script type="text/html" id="barDemo">
            <a class="layui-btn layui-btn-xs" lay-event="detail">查看</a>
```

```
            <a class="layui-btn layui-btn-xs" lay-event="edit">编辑</a>
            <a class="layui-btn layui-btn-danger layui-btn-xs"
lay-event="del">删除</a>
        </script>

        <script>
            var table = layui.table;
            //第一个实例
            table.render({
                id: "mytable",
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
                            type: "checkbox",
                        },
                        {
                            field: 'id',
                            title: 'ID',
                        }, {
                            field: 'username',
                            title: '账号',
                        }, {
                            field: 'password',
                            title: '账号',
                        }
                    ]
                ]
            });

            function changeSize(){
                $("body div:first").width("500px");
            }
            function resize(){
                var table = layui.table;
                table.resize("mytable");
            }
        </script>
        <input type="button" onclick="javascript:changeSize()" value="改
变大小" />
        <br />
        <input type="button" onclick="javascript:resize()" value="resize"
```

```
/>
    </body>
</html>
```

1.23.9 表格重载

当表格中的数据被新的数据所替代而又不想重新渲染表格时可以使用表格重载 reload 方法。有两种方式实现表格重载，如图 1-xx 所示。

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 参数 *options* 即为各项基础参数	所有渲染方式
tableIns.reload(options)	参数同上 tableIns 可通过 var tableIns = table.render() 得到	仅限方法级渲染

但 reload()方法在复杂的场景下，Layui 官方并没有做好相关的功能处理，所以很多使用者在使用 reload()方法时会出现一些 bug，如图 1-xx 所示。

建议大家还是使用 render()方法重新对表格进行整体的渲染与数据处理，减少莫名其妙的 bug，另外 Layui 官方已经很久没有更新该框架，现有的 bug 或隐式的 bug 估计还会在不同的场景下出现，所以尽量不要以身试坑，使用 render()方法就完全避免了这些 bug 和坑。

自动渲染的示例如图 1-xx 所示。

示例1：自动化渲染的重载

```
【HTML】
<table class="layui-table" lay-data="{id: 'idTest'}"> …… </table>

【JS】
table.reload('idTest', {
  url: '/api/table/search'
  ,where: {} //设定异步数据接口的额外参数
  //,height: 300
});

```

方法渲染的示例如图 1-xx 所示。

示例2：方法级渲染的重载

```
//所获得的 tableIns 即为当前容器的实例
var tableIns = table.render({
  elem: '#id'
  ,cols: [] //设置表头
  ,url: '/api/data' //设置异步接口
  ,id: 'idTest'
});

//这里以搜索为例
tableIns.reload({
  where: { //设定异步数据接口的额外参数，任意设
    aaaaaa: 'xxx'
    ,bbb: 'yyy'
    //…
  }
  ,page: {
    curr: 1 //重新从第 1 页开始
  }
});
//上述方法等价于
table.reload('idTest', {
  where: { //设定异步数据接口的额外参数，任意设
    aaaaaa: 'xxx'
    ,bbb: 'yyy'
    //…
  }
  ,page: {
    curr: 1 //重新从第 1 页开始
  }
}); //只重载数据

```

1.23.10 导出任意数据

尽管 table 的工具栏内置了数据导出按钮，但有时可能需要通过方法去导出任意列中的数据，那么可以借助以下方法：

语法：table.exportFile(id, data, type)。

示例代码如图 1-xx 所示。

```
code
var ins1 = table.render({
  elem: '#demo'
  ,id: 'test'
  //,…… //其它参数
})

//将上述表格示例导出为 csv 文件
table.exportFile(ins1.config.id, data); //data 为该实例中的任意数量的数据

```

事实上，该方法也可以不用依赖 table 的实例，可直接导出任意数据：

```
code
table.exportFile(['名字','性别','年龄'], [
  ['张三','男','20'],
  ['李四','女','18'],
  ['王五','女','19']
], 'csv'); //默认导出 csv，也可以为：xls

```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            var mytable=table.render({
                elem: '#demo',
```

```
            url: 'new.html', //数据接口
            page: true, //开启分页
            cols: [
                [ //表头
                    {
                        field: 'id',
                        title: 'ID',
                        width: 200,
                        sort: true,
                        fixed: 'left'
                    }, {
                        field: 'username',
                        title: '账号',
                        width: 200
                    }, {
                        field: 'password',
                        title: '账号',
                        width: 200,
                        sort: true
                    }
                ]
            ]
        });

        $(document).ready(function (){
            $("#exportButton1").click(function (){
                var table = layui.table;
                //将上述表格示例导出为csv文件
                table.exportFile(mytable.config.id, mytable.data);
//data为该实例中的任意数量的数据
            });

            $("#exportButton2").click(function (){
                var table = layui.table;
                //将上述表格示例导出为xls文件
                table.exportFile(mytable.config.id,
mytable.data,"xls"); //data为该实例中的任意数量的数据
            });

            $("#exportButton3").click(function (){
                table.exportFile(['名字','性别','年龄'], [
                  ['张三','男','20'],
                  ['李四','女','18'],
                  ['王五','女','19']
```

```
                ], 'xls'); //默认导出 csv，也可以为：xls
            });
        });
    </script>
    <input type="button" value="导出cvs" id="exportButton1" />
    <br />
    <input type="button" value="导出xls" id="exportButton2" />
    <br />
    <input type="button" value="导出xls" id="exportButton3" />
</body>
</html>
```

1.23.11 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，示例代码如图 1-xx 所示。

```
code
01. 原始容器
02. <table id="demo" lay-filter="test"></table>
03. 
04. 工具栏模板：
05. <script type="text/html" id="toolbarDemo">
06.   <div class="layui-btn-container">
07.     <button class="layui-btn layui-btn-sm" lay-event="add">添加</button>
08.     <button class="layui-btn layui-btn-sm" lay-event="delete">删除</button>
09.     <button class="layui-btn layui-btn-sm" lay-event="update">编辑</button>
10.   </div>
11. </script>
12. 
13. <script;>
14. //JS 调用：
15. table.render({
16.   elem: '#demo'
17.   ,toolbar: '#toolbarDemo'
18.   //,…… //其他参数
19. });
20. 
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>
37. 
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
```

```
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="mydiv" style="display: none;">
            <button class="layui-btn layui-btn-sm" lay-event="add">增加
</button>
            <button class="layui-btn layui-btn-sm" lay-event="delete">删除
</button>
            <button class="layui-btn layui-btn-sm" lay-event="update">修改
</button>
        </div>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            var mytable = table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                toolbar: "#mydiv",
                cols: [
                    [ //表头
                        {
                            field: 'id',
                            title: 'ID',
                            width: 200,
                            sort: true,
                            fixed: 'left'
                        }, {
                            field: 'username',
                            title: '账号',
                            width: 200
                        }, {
                            field: 'password',
                            title: '账号',
                            width: 200,
                            sort: true
                        }
                    ]
                ]
            });
            //监听事件
            table.on('toolbar(test)', function(obj) {
                var checkStatus = table.checkStatus(obj.config.id);
```

```
                switch (obj.event) {
                    case 'add':
                        layer.msg('添加');
                        break;
                    case 'delete':
                        layer.msg('删除');
                        break;
                    case 'update':
                        layer.msg('编辑');
                        break;
                };
            });
        </script>
    </body>
</html>
```

1.23.12 监听行中 checkbox 状态的改变

点击复选框时触发，回调函数返回一个 object 参数，携带的成员如下：

```
table.on('checkbox(test)', function(obj){
  console.log(obj.checked); //当前是否选中状态
  console.log(obj.data); //选中行的相关数据
  console.log(obj.type); //如果触发的是全选，则为：all，如果触发的是单选，则为：one
});
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            var mytable = table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
```

```
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
                            field: '',
                            title: '',
                            type: "checkbox"
                        },
                        {
                            field: 'id',
                            title: 'ID',
                            width: 200,
                        }, {
                            field: 'username',
                            title: '账号',
                            width: 200
                        }, {
                            field: 'password',
                            title: '账号',
                            width: 200,
                        }
                    ]
                ]
            });
            table.on('checkbox(test)', function(obj) {
                console.log(obj.checked); //当前是否选中状态
                console.log(obj.data); //选中行的相关数据
                console.log(obj.type); //如果触发的是全选，则为：all，如果触发的是单选，则为：one
            });
        </script>
    </body>
</html>
```

1.23.13 监听单元格编辑

单元格被编辑且值发生改变时触发，回调函数返回一个 object 参数，携带的成员如下：
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"
  console.log(obj.value); //得到修改后的值
  console.log(obj.field); //当前编辑的字段名
  console.log(obj.data); //所在行的所有相关数据
});

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            var mytable = table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
                page: true, //开启分页
                cols: [
                    [ //表头
                        {
                            field: '',
                            title: '',
                            type: "checkbox"
                        },
                        {
                            field: 'id',
                            title: 'ID',
                            width: 200
                        }, {
                            field: 'username',
                            title: '账号',
                            width: 200,
                            edit: true
                        }, {
                            field: 'password',
                            title: '账号',
                            width: 200,
                        }
                    ]
                ]
            });
```

```
            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"
                console.log(obj.value); //得到修改后的值
                console.log(obj.field); //当前编辑的字段名
                console.log(obj.data); //所在行的所有相关数据
            });
        </script>
    </body>
</html>
```

1.23.14 监听行单双击事件

点击或双击行时触发。

示例代码如下：

```
//监听行单击事件
table.on('row(test)', function(obj){
  console.log(obj.tr) //得到当前行元素对象
  console.log(obj.data) //得到当前行数据
  //obj.del(); //删除当前行
  //obj.update(fields) //修改当前行数据
});

//监听行双击事件
table.on('rowDouble(test)', function(obj){
  //obj 同上
});
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            var mytable = table.render({
```

```
            elem: '#demo',
            url: 'new.html', //数据接口
            page: true, //开启分页
            cols: [
                [ //表头
                    {
                        field: '',
                        title: '',
                        type: "checkbox"
                    },
                    {
                        field: 'id',
                        title: 'ID',
                        width: 200
                    }, {
                        field: 'username',
                        title: '账号',
                        width: 200
                    }, {
                        field: 'password',
                        title: '账号',
                        width: 200,
                    }
                ]
            ]
        });
        //监听行单击事件
        table.on('row(test)', function(obj) {
            console.log("单击");
            console.log(obj.tr) //得到当前行元素对象
            console.log(obj.data) //得到当前行数据
            //obj.del(); //删除当前行
            //obj.update(fields) //修改当前行数据
        });

        //监听行双击事件
        table.on('rowDouble(test)', function(obj) {
            console.log("双击");
            console.log(obj.tr) //得到当前行元素对象
            console.log(obj.data) //得到当前行数据
        });
    </script>
</body>
</html>
```

1.23.15 监听排序切换

点击表头排序时触发，它通过在基础参数中设置 autoSort: false 时使用，以完成服务端的排序，而不是默认的前端排序。该事件的回调函数返回一个 object 参数，携带的成员如下：

```
//禁用前端自动排序，以便由服务端直接返回排序好的数据
table.render({
  elem: '#id'
  ,autoSort: false //禁用前端自动排序
  //,… //其它参数省略
});

//监听排序事件
table.on('sort(test)', function(obj){ //注：sort 是工具条事件名，test 是 table 原始容器的属性 lay-filter="对应的值"
  console.log(obj.field); //当前排序的字段名
  console.log(obj.type); //当前排序类型：desc（降序）、asc（升序）、null（空对象，默认排序）
  console.log(this); //当前排序的 th 对象

  //尽管 table 自带排序功能，但并没有请求服务端。
  //有些时候，可能需要根据当前排序的字段，重新向服务端发送请求，从而实现服务端排序。
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <table id="demo" lay-filter="test"></table>
        <script>
            var table = layui.table;
            //第一个实例
            var mytable = table.render({
                elem: '#demo',
                url: 'new.html', //数据接口
```

```
                page: true, //开启分页
                autoSort: false,
                cols: [
                    [ //表头
                        {
                            field: '',
                            title: '',
                            type: "checkbox"
                        },
                        {
                            field: 'id',
                            title: 'ID',
                            width: 200,
                            sort: true
                        }, {
                            field: 'username',
                            title: '账号',
                            width: 200
                        }, {
                            field: 'password',
                            title: '账号',
                            width: 200,
                        }
                    ]
                ]
            });
            //监听排序事件
            table.on('sort(test)', function(obj) { //注：sort是工具条事件名，test是table原始容器的属性 lay-filter="对应的值"
                console.log(obj.field); //当前排序的字段名
                console.log(obj.type); //当前排序类型：desc（降序）、asc（升序）、null（空对象，默认排序）
                console.log(this); //当前排序的th对象
                //尽管table自带排序功能，但并没有请求服务端。
                //有些时候，可能需要根据当前排序的字段，重新向服务端发送请求，从而实现服务端排序。
            });
        </script>
    </body>
</html>
```

1.24 表单模块 form

模块加载名称：form。

1.24.1 表单基本使用

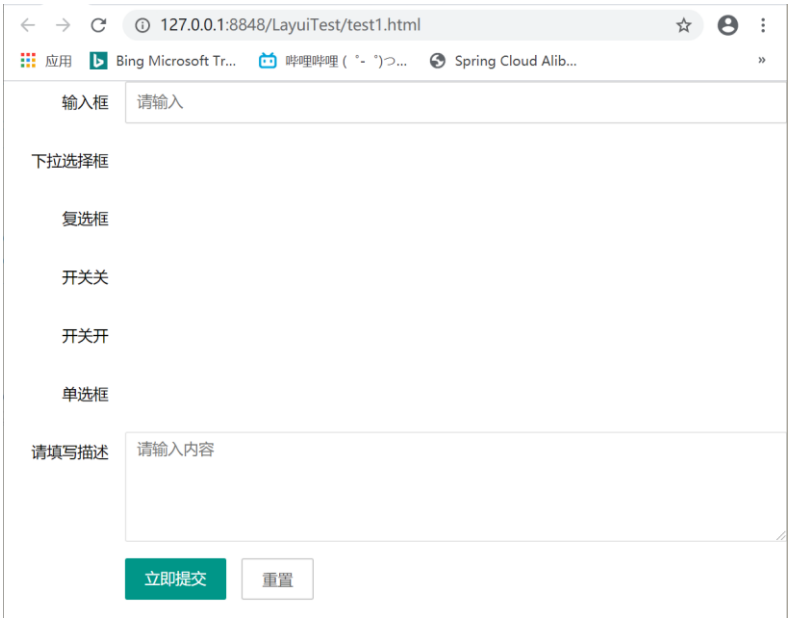
Layui 针对各种表单元素做了较为全面的 UI 支持，无需去书写那些 UI 结构，只需要写 HTML 原始的 input、select、textarea 等等这些基本的标签即可。在 UI 上的渲染只要求一点：必须给表单体系所在的父元素加上 class="layui-form"，一切的工作都会在加载完 form 模块后自动完成，示例代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <form class="layui-form">
            <!-- 提示：如果你不想用form，你可以换成div等任何一个普通元素 -->
            <div class="layui-form-item">
                <label class="layui-form-label">输入框</label>
                <div class="layui-input-block">
                    <input type="text" name="" placeholder="请输入"
autocomplete="off" class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">下拉选择框</label>
                <div class="layui-input-block">
                    <select name="interest" lay-filter="aihao">
                        <option value="0">写作</option>
                        <option value="1">阅读</option>
                    </select>
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">复选框</label>
                <div class="layui-input-block">
                    <input type="checkbox" name="like[write]" title="写作">
                    <input type="checkbox" name="like[read]" title="阅读">
```

```
            </div>
        </div>
        <div class="layui-form-item">
            <label class="layui-form-label">开关关</label>
            <div class="layui-input-block">
                <input type="checkbox" lay-skin="switch">
            </div>
        </div>
        <div class="layui-form-item">
            <label class="layui-form-label">开关开</label>
            <div class="layui-input-block">
                <input type="checkbox" checked lay-skin="switch">
            </div>
        </div>
        <div class="layui-form-item">
            <label class="layui-form-label">单选框</label>
            <div class="layui-input-block">
                <input type="radio" name="sex" value="0" title="男">
                <input type="radio" name="sex" value="1" title="女"
checked>
            </div>
        </div>
        <div class="layui-form-item layui-form-text">
            <label class="layui-form-label">请填写描述</label>
            <div class="layui-input-block">
                <textarea placeholder="请输入内容"
class="layui-textarea"></textarea>
            </div>
        </div>
        <div class="layui-form-item">
            <div class="layui-input-block">
                <button class="layui-btn" lay-submit lay-filter="*">立
即提交</button>
                <button type="reset" class="layui-btn
layui-btn-primary">重置</button>
            </div>
        </div>
    </form>
    <script>
        layui.use('form', function() {
            var form = layui.form;
            //各种基于事件的操作，下面会有进一步介绍
        });
    </script>
```

```
    </body>
</html>
```

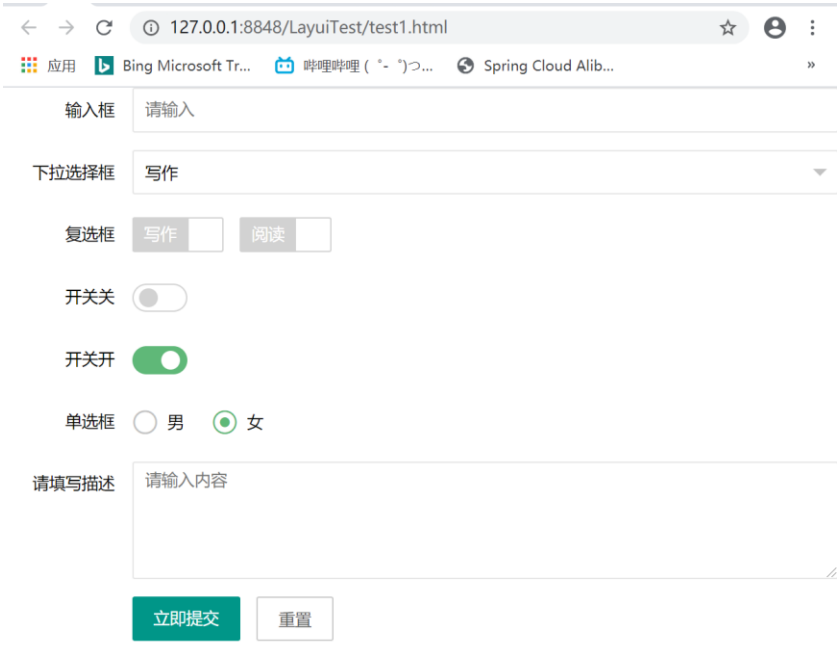
运行效果如图 1-xx 所示。



部分表单没有渲染出来，更改 javascript 代码，执行 render()方法，示例代码如下：

```
    <script>
        layui.use('form', function() {
            var form = layui.form;
            form.render();
            //各种基于事件的操作，下面会有进一步介绍
        });
    </script>
```

运行效果如图 1-xx 所示。



成功渲染出表单。

1.24.2 更新渲染

有些时候，表单元素可能是动态创建并添加的，这时 form 模块的自动化渲染是失效的，这时只需要执行 form.render(type, filter); 方法即可。

第一个参数：type，为表单的 type 类型，可选。默认对全部类型的表单进行一次渲染更新，可局部刷新的 type 值如图 1-xx 所示。

参数（type）值	描述
select	刷新select选择框渲染
checkbox	刷新checkbox复选框（含开关）渲染
radio	刷新radio单选框框渲染

注意：想要 UI 被成功渲染，必须给表单所在的父元素加上 class="layui-form"类样式。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>layui.form小例子</title>
        <link rel="stylesheet" href="layui/css/layui.css" media="all">
        <script src="js/jquery3.5.1.js"></script>
```

```
    </head>
    <body>
        <form class="layui-form">
            <input type="text" class="layui-input">
            <select name="interest">
                <option value="0">写作</option>
                <option value="1">阅读</option>
            </select>
            <input type="checkbox" name="like[write]">
            <input type="checkbox" name="like[read]">
            <input type="radio" name="sex" value="0">
            <input type="radio" name="sex" value="1" checked>
            <button class="layui-btn" lay-submit>立即提交</button>
            <button type="reset" class="layui-btn layui-btn-primary">重置
</button>
        </form>
        <br />
        <br />
        <br />
        <br />
        <br />
        <br />
        <div id="newDiv" class="layui-form"></div>
        <script src="layui/layui.all.js"></script>
        <script>
            var form = layui.form;

            $(document).ready(function () {
                $("#createNewElement").click(function () {
                    var newText = $("<input type='text'
class='layui-input'>");
                    $("#newDiv").append(newText);

                    $("#newDiv").append("<br/>");

                    var newSelect = $("<select></select>");
                    var newOption1 = $("<option></option>");
                    $(newOption1).val(1);
                    $(newOption1).text(1);
                    var newOption2 = $("<option></option>");
                    $(newOption2).val(2);
                    $(newOption2).text(2);
                    $(newSelect).append(newOption1);
                    $(newSelect).append(newOption2);
```

```
                $("#newDiv").append(newSelect);
                form.render("select");

                $("#newDiv").append("<br/>");

                var newCheckbox1 = $("<input type='checkbox'
class='layui-input' lay-skin='primary'>");
                var newCheckbox2 = $("<input type='checkbox'
class='layui-input' lay-skin='primary'>");
                $("#newDiv").append(newCheckbox1);
                $("#newDiv").append(newCheckbox2);
                form.render("checkbox");

                $("#newDiv").append("<br/>");

                var newRadio1 = $("<input type='radio' name='sex'
class='layui-input'>");
                var newRadio2 = $("<input type='radio' name='sex'
class='layui-input'>");
                $("#newDiv").append(newRadio1);
                $("#newDiv").append(newRadio2);
                form.render("radio");
            });

        });
    </script>
    <input type="button" value="创建新的元素并渲染"
id="createNewElement">
</body>
</html>
```

第二个参数：filter，为 class="layui-form"所在元素的 lay-filter=""的值，可以借助该参数对表单完成局部更新，示例代码如下：

【HTML】

```
<div class="layui-form" lay-filter="test1">
  ...
</div>

<div class="layui-form" lay-filter="test2">
  ...
</div>
```

【JavaScript】

```
form.render(null, 'test1'); //更新 lay-filter="test1"所在容器内的全部表单状态
```

form.render('select', 'test2'); //更新 lay-filter="test2"所在容器内的全部 select 状态
//……

1.24.3 预设元素属性

如图 1-xx 所示的基础属性可以应用在标签上。

属性名	属性值	说明
title	任意字符	设定元素名称，一般用于checkbox、radio框
lay-skin	switch（开关风格） primary（原始风格）	定义元素的风格，目前只对 *checkbox* 元素有效，可将其转变为开关样式
lay-ignore	任意字符或不设值	是否忽略元素美化处理。设置后，将不会对该元素进行初始化渲染，即保留系统风格
lay-filter	任意字符	事件过滤器，主要用于事件的精确匹配，跟选择器是比较类似的。其实它并不私属于form模块，它在 layui 的很多基于事件的接口中都会应用到。
lay-verify	required（必填项） phone（手机号） email（邮箱） url（网址） number（数字） date（日期） identity（身份证） 自定义值	同时支持多条规则的验证，格式： *lay-verify="验证A|验证B"* 如： *lay-verify="required|phone|number"* 另外，除了我们内置的校验规则，你还可以给他设定任意的值，比如lay-verify="pass"，那么你就需要借助form.verify()方法对pass进行一个校验规则的定义。详见表单验证
lay-verType	tips（吸附层） alert（对话框） msg（默认）	用于定义异常提示层模式。
lay-reqText	任意字符	用于自定义必填项（即设定了 lay-verify="required" 的表单）的提示文本 注意： *该功能为 layui 2.5.0 新增*
lay-submit	无需填写值	绑定触发提交的元素，如button

标签应用基础属性的示例代码如下：

```
<input type="text" lay-verify="email">
<input type="checkbox" checked lay-skin="switch" lay-filter="encrypt" title="是否加密">
<button lay-submit>提交</button>
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <form class="layui-form">
            <div class="layui-form-item">
                <label class="layui-form-label">账号</label>
```

```
                <div class="layui-input-block">
                    <input type="text" name="username" placeholder="请输入"
autocomplete="off" class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">账号</label>
                <div class="layui-input-block">
                    <input type="text" name="username" placeholder="请输入"
autocomplete="off" class="layui-input" lay-verify="required"
                     lay-verType="tips" lay-reqText="账号不能为空!">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">开关</label>
                <div class="layui-input-block">
                    <input type="checkbox" lay-skin="switch" checked>
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">复选框</label>
                <div class="layui-input-block">
                    <input type="checkbox" lay-skin="primary"
name="like[write]" title="写作">
                    <input type="checkbox" lay-skin="primary"
name="like[read]" title="阅读">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">单选框</label>
                <div class="layui-input-block">
                    <input type="radio" name="sex" value="0" title="男">
                    <input type="radio" name="sex" value="1" title="女"
checked>
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">原始复选框</label>
                <div class="layui-input-block">
                    <input type="checkbox" lay-skin="primary"
name="like[write]" title="写作" lay-ignore />
                    <input type="checkbox" lay-skin="primary"
name="like[read]" title="阅读" lay-ignore />
                </div>
```

```
            </div>
            <div class="layui-form-item">
                <div class="layui-input-block">
                    <button class="layui-btn" lay-submit lay-filter="*">立
即提交</button>
                    <button type="reset" class="layui-btn
layui-btn-primary">重置</button>
                </div>
            </div>
        </form>
        <script>
            layui.use('form', function() {
                var form = layui.form;
                form.render();
                //各种基于事件的操作，下面会有进一步介绍
            });
        </script>
    </body>
</html>
```

1.24.4 事件监听

语法：form.on('event(过滤器值)', callback);

form 模块在 Layui 事件机制中注册了专属事件，所以当使用 layui.onevent()自定义模块事件时，请勿占用 form 默认事件名，form 默认支持的事件名如图 1-xx 所示。

event	描述
select	监听select下拉选择事件
checkbox	监听checkbox复选框勾选事件
switch	监听checkbox复选框开关事件
radio	监听radio单选框事件
submit	监听表单提交事件

默认情况下，事件所监听的是全部的 form 模块元素，但如果只想监听某一个元素，使用事件过滤器即可，示例代码如下：

<select lay-filter="test"></select>

```
form.on('select(test)', function(data){
  console.log(data);
});
```

1.24.5 监听 select 选择

下拉选择框被选中时触发，回调函数返回一个 object 参数，携带两个成员，示例代码如图 1-xx 所示。

语法

```
form.on('select(filter)', function(data){
  console.log(data.elem); //得到select原始DOM对象
  console.log(data.value); //得到被选中的值
  console.log(data.othis); //得到美化后的DOM对象
});
```

请注意：如果想监听所有的 select，去掉过滤器(filter)即可。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-form">
            <label class="layui-form-label">下拉选择框</label>
            <div class="layui-input-block">
                <select name="interest" lay-filter="aihao">
                    <option value="0">写作</option>
                    <option value="1">阅读</option>
                </select>
            </div>
        </div>
        <script>
            var form = layui.form;
```

```
            form.render();
            form.on('select(aihao)', function(data) {
                console.log(data.elem); //得到select原始DOM对象
                console.log(data.value); //得到被选中的值
                console.log(data.othis); //得到美化后的DOM对象
            });
        </script>
    </body>
</html>
```

1.24.6 监听 checkbox 复选

复选框点击勾选时触发，回调函数返回一个 object 参数，携带两个成员，测试代码如图 1-xx 所示。

语法

```
form.on('checkbox(filter)', function(data){
  console.log(data.elem); //得到checkbox原始DOM对象
  console.log(data.elem.checked); //是否被选中，true或者false
  console.log(data.value); //复选框value值，也可以通过data.elem.value得到
  console.log(data.othis); //得到美化后的DOM对象
});

```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-form">
            <label class="layui-form-label">复选框</label>
            <div class="layui-input-block">
                <input type="checkbox" lay-skin="primary"
name="like[write]" title="写作" lay-filter="sameGroup" value="写作">
                <input type="checkbox" lay-skin="primary" name="like[read]"
title="阅读" lay-filter="sameGroup" value="阅读">
```

```
            </div>
        </div>
        <script>
            var form = layui.form;
            form.render();
            form.on('checkbox(sameGroup)', function(data) {
                console.log(data.elem); //得到select原始DOM对象
                console.log(data.value); //得到被选中的值
                console.log(data.othis); //得到美化后的DOM对象
            });
        </script>
    </body>
</html>
```

1.24.7 监听 switch 开关

开关被点击时触发，回调函数返回一个 object 参数，携带两个成员，测试代码如图 1-xx 所示。

语法

```
form.on('switch(filter)', function(data){
  console.log(data.elem); //得到checkbox原始DOM对象
  console.log(data.elem.checked); //开关是否开启，true或者false
  console.log(data.value); //开关value值，也可以通过data.elem.value得到
  console.log(data.othis); //得到美化后的DOM对象
});

```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-form">
            <label class="layui-form-label">开关</label>
            <div class="layui-input-block">
```

```
            <input type="checkbox" lay-skin="switch" checked
lay-filter="myswitch">
        </div>
    </div>
    <script>
        var form = layui.form;
        form.render();
        form.on('switch(myswitch)', function(data) {
            console.log(data.elem); //得到checkbox原始DOM对象
            console.log(data.elem.checked); //开关是否开启，true或者
false
            console.log(data.value); //开关value值，也可以通过
data.elem.value得到
            console.log(data.othis); //得到美化后的DOM对象
        });
    </script>
</body>
</html>
```

1.24.8 监听 radio 单选

radio 单选框被点击时触发，回调函数返回一个 object 参数，携带两个成员，示例代码如图 1-xx 所示。

语法

```
form.on('radio(filter)', function(data){
  console.log(data.elem); //得到radio原始DOM对象
  console.log(data.value); //被点击的radio的value值
});
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
```

```
    </head>
    <body>
        <div class="layui-form">
            <label class="layui-form-label">单选框</label>
            <div class="layui-input-block">
                <input type="radio" name="sex" value="0" title="男"
lay-filter="sex">
                <input type="radio" name="sex" value="1" title="女" checked
lay-filter="sex">
            </div>
        </div>
        <script>
            var form = layui.form;
            form.render();
            form.on('radio(sex)', function(data) {
                console.log(data.elem); //得到radio原始DOM对象
                console.log(data.value); //被点击的radio的value值
            });
        </script>
    </body>
</html>
```

1.24.9 监听 submit 提交

按钮点击或者表单被执行提交时触发，其中回调函数只有在验证全部通过后才会进入，回调返回三个成员，示例代码如图 1-xx 所示。

语法

```
form.on('submit(*)', function(data){
  console.log(data.elem) //被执行事件的元素DOM对象，一般为button对象
  console.log(data.form) //被执行提交的form对象，一般在存在form标签时才会返回
  console.log(data.field) //当前容器的全部表单字段，名值对形式：{name: value}
  return false; //阻止表单跳转。如果需要表单跳转，去掉这段即可。
});

```

温馨提示：上述的 submit(*)中的*星号为事件过滤器的值，示例代码如图 1-xx 所示。

code

```
<button lay-submit lay-filter="*">提交</button>

```

可以把*号换成任意的值，如：lay-filter="go"，但监听事件时也要改成：
form.on('submit(go)', callback);

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <form action="http://www.baidu.com">
            <div class="layui-form-item">
                <div class="layui-input-block">
                    <button class="layui-btn" lay-submit lay-filter="*">立
即提交</button>
                    <button type="reset" class="layui-btn
layui-btn-primary">重置</button>
                </div>
            </div>
        </form>
        <script>
            var form = layui.form;
            form.render();
            form.on('submit(*)', function(data) {
                console.log(data.elem) //被执行事件的元素DOM对象，一般为
button对象
                console.log(data.form) //被执行提交的form对象，一般在存在form
标签时才会返回
                console.log(data.field) //当前容器的全部表单字段，名值对形式：
{name: value}
                return false; //阻止表单跳转。如果需要表单跳转，去掉这段即可。
            });
        </script>
    </body>
</html>
```

1.24.10 表单赋值/取值

语法：form.val('filter', object);

用于给指定表单集合的元素赋值和取值。如果 object 参数存在，则为赋值；如果 object 参数不存在，则为取值。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <form class="layui-form" lay-filter="formTest">
            <div class="layui-form-item">
                <label class="layui-form-label">账号</label>
                <div class="layui-input-block">
                    <input type="text" name="username" placeholder="请输入"
autocomplete="off" class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">密码</label>
                <div class="layui-input-block">
                    <input type="text" name="password" placeholder="请输入"
autocomplete="off" class="layui-input">
                </div>
            </div>
            <div class="layui-form-item layui-form-text">
                <label class="layui-form-label">请填写描述</label>
                <div class="layui-input-block">
                    <textarea placeholder="请输入内容" name="bigtext"
class="layui-textarea"></textarea>
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">单选框</label>
                <div class="layui-input-block">
                    <input type="radio" name="sex" value="1" title="男">
                    <input type="radio" name="sex" value="0" title="女">
                </div>
            </div>
            <div class="layui-form-item">
```

```
            <label class="layui-form-label">开关</label>
            <div class="layui-input-block">
                <input type="checkbox" lay-skin="switch"
name="myswitch">
            </div>
        </div>
        <div class="layui-form-item">
            <label class="layui-form-label">下拉选择框</label>
            <div class="layui-input-block">
                <select name="interest" lay-filter="aihao">
                    <option value="0">写作</option>
                    <option value="1">阅读</option>
                </select>
            </div>
        </div>
        <div class="layui-form-item">
            <label class="layui-form-label">复选框</label>
            <div class="layui-input-block">
                <input type="checkbox" lay-skin="primary"
name="like[write]" title="写作">
                <input type="checkbox" lay-skin="primary"
name="like[read]" title="阅读">
            </div>
        </div>

        <div class="layui-form-item">
            <div class="layui-input-block">
                <button type="button" class="layui-btn"
id="setButton">set</button>
                <button type="button" class="layui-btn"
id="getButton">get</button>
            </div>
        </div>
    </form>
    <script>
        var form = layui.form;
        layui.use('form', function() {
            var form = layui.form;
            form.render();
            //各种基于事件的操作，下面会有进一步介绍
        });

        $(document).ready(function() {
            $("#setButton").click(function() {
```

```
                //formTest即class="layui-form"所在元素属性lay-filter=""
对应的值
                form.val("formTest", {
                    "username": "我是账号"
                });
                form.val("formTest", {
                    "password": "我是密码"
                });
                form.val("formTest", {
                    "bigtext": "我是大文本"
                });
                form.val("formTest", {
                    "sex": "0"
                });
                form.val("formTest", {
                    "myswitch": true
                });
                form.val("formTest", {
                    "interest": 1
                });
                form.val("formTest", {
                    "like[write]": true
                });
            });

            $("#getButton").click(function() {
                //获取表单区域所有值
                var data1 = form.val("formTest");
                alert(data1);
            });

        });
    </script>
</body>
</html>
```

1.24.11 表单验证

Layui 对表单的验证进行了非常巧妙的支持，大多数时候只需要在表单元素上加上lay-verify=""属性值即可，示例代码如图 1-xx 所示。

code

```
<input type="text" lay-verify="email">

还同时支持多条规则的验证，如下：
<input type="text" lay-verify="required|phone|number">

```

上述对输入框定义了一个邮箱规则的校验，它会在 form 模块内部完成。

除了内置的校验规则外，还可以自定义验证规则，通常对于比较复杂的校验，这是非常有必要的，示例代码如图 1-xx 所示。

语法

```
form.verify({
  username: function(value, item){ // value：表单的值、item：表单的DOM对象
    if(!new RegExp("^[a-zA-Z0-9_\u4e00-\u9fa5\\s·]+$").test(value)){
      return '用户名不能有特殊字符';
    }
    if(/(^\_)|(\__)|(\_+$)/.test(value)){
      return '用户名首尾不能出现下划线\'_\'';
    }
    if(/^\d+\d+\d$/.test(value)){
      return '用户名不能全为数字';
    }
  }

  //我们既支持上述函数式的方式，也支持下述数组的形式
  //数组的两个值分别代表：[正则匹配、匹配不符时的提示文字]
  ,pass: [
    /^[\S]{6,12}$/
    ,'密码必须6到12位，且不能出现空格'
  ]
});

```

当自定义了类似上面的验证规则后，只需要把 key 赋值给输入框的 lay-verify 属性即可，示例代码如图 1-xx 所示。

code

```
<input type="text" lay-verify="username" placeholder="请输入用户名">
<input type="password" lay-verify="pass" placeholder="请输入密码">

```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <form class="layui-form" action="http://ww.baidu.com">
            <div class="layui-form-item">
                <label class="layui-form-label">账号</label>
                <div class="layui-input-block">
                    <input type="text" name="username" placeholder="请输入"
autocomplete="off" class="layui-input"
lay-verify="required|isExistsUsername">
                </div>
            </div>
            <div class="layui-form-item">
                <div class="layui-input-block">
                    <button class="layui-btn" lay-submit>立即提交</button>
                </div>
            </div>
        </form>
        <script>
            var form = layui.form;
            form.verify({
                isExistsUsername: function(value, item) { //value：表单的值、
item：表单的DOM对象
                    if (value == '中国') {
                        return '用户名"中国"已经存在，不能重复注册！';
                    }
                }
            });
        </script>
    </body>
</html>
```

1.25 上传 upload

上传模块自 Layui2.0 的版本开始，进行了全面重写，这使得它的功能不再那么单一，它所包含的不仅是更为强劲的功能，还有灵活的 UI。任何元素都可以作为上传组件来调用，譬如按钮、图片、普通的 DIV 等等，而不再是一个单调的 file 文件域。

模块加载名称：upload。

1.25.1 前端界面效果

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
        </button>
        <script>
            var upload = layui.upload;
            //执行实例
            var uploadInst = upload.render({
                elem: '#test1', //绑定元素
                url: '/upload/', //上传接口
                done: function(res) {
                    //上传完毕回调
                    alert("成功上传");
                },
                error: function() {
                    //请求异常回调
                    alert("上传失败");
                }
            });
        </script>
    </body>
</html>
```

以上代码只是演示上传的前端界面，并没有后台支持，所以上传失败。

1.25.2 核心方法与基础参数选项

使用 upload 模块必须与 upload.render(options)方法打交道，其中的 options 即为基础参数，它是一个对象，示例代码如图 1-xx 所示。

code

```
var upload = layui.upload; //得到 upload 对象

//创建一个上传组件
upload.render({
  elem: '#id'
  ,url: ''
  ,done: function(res, index, upload){ //上传后的回调

  }
  //,accept: 'file' //允许上传的文件类型
  //,size: 50 //最大允许上传的文件大小
  //,……
})
```

从 Layui 2.1.0 开始，允许直接在元素上设定基础参数，示例代码如图 1-xx 所示。

code

```
【HTML】
<button class="layui-btn test" lay-data="{url: '/a/'}">上传图片</button>
<button class="layui-btn test" lay-data="{url: '/b/', accept: 'file'}">上传文件</button>

【JS】
upload.render({
  elem: '.test'
  ,done: function(res, index, upload){
    //获取当前触发上传的元素，一般用于 elem 绑定 class 的情况，注意：此乃 layui 2.1.0 新增
    var item = this.item;
  }
})
```

更多参数如图 1-xx 所示。

参数选项	说明	类型	默认值
elem	指向容器选择器，如：elem: '#id'。也可以是DOM对象	string/object	-
url	服务端上传接口，返回的数据规范请详见下文	string	-

data	请求上传接口的额外参数。如：data: {id: 'xxx'}
	从 layui 2.2.6 开始，支持动态值，如:
	code layui.code
	01. data: {
	02. id: function(){
	03. return $('#id').val();
	04. }
	05. }
	06.
		object	-

headers	接口的请求头。如：*headers: {token: 'sasasas'}*。注：该参数为 layui 2.2.6 开始新增		
accept	指定允许上传时校验的文件类型，可选值有：*images*（图片）、*file*（所有文件）、*video*（视频）、*audio*（音频）	string	images
acceptMime	规定打开文件选择框时，筛选出的文件类型，值为用逗号隔开的 MIME 类型列表。如： *acceptMime: 'image/*'*（只显示图片文件） *acceptMime: 'image/jpg, image/png'*（只显示 jpg 和 png 文件） 注：该参数为 layui 2.2.6 开始新增	string	images
exts	允许上传的文件后缀。一般结合 *accept* 参数类设定。假设 accept 为 file 类型时，那么你设置 *exts: 'zip|rar|7z'* 即代表只允许上传压缩格式的文件。如果 accept 未设定，那么限制的就是图片的文件格式	string	jpg|png|gif|bmp|jpeg
auto	是否选完文件后自动上传。如果设定 *false*，那么需要设置 *bindAction* 参数来指向一个其它按钮提交上传	boolean	true
bindAction	指向一个按钮触发上传，一般配合 auto: false 来使用。值为选择器或 DOM对象，如：bindAction: '#btn'	string/object	-
field	设定文件域的字段名	string	file
size	设置文件最大可允许上传的大小，单位 KB。不支持ie8/9	number	0（即不限制）
multiple	是否允许多文件上传。设置 *true* 即可开启。不支持ie8/9	boolean	false
number	设置同时可上传的文件数量，一般配合 multiple 参数出现。 注意：*该参数为 layui 2.2.3 开始新增*	number	0（即不限制）
drag	是否接受拖拽的文件上传，设置 *false* 可禁用。不支持ie8/9	boolean	true

回调			
choose	选择文件后的回调函数。返回一个object参数，详见下文	function	-
before	文件提交上传前的回调。返回一个object参数（同上），详见下文	function	-
done	执行上传请求后的回调。返回三个参数，分别为：*res*（服务端响应信息）、*index*（当前文件的索引）、*upload*（重新上传的方法，一般在文件上传失败后使用）。详见下文	function	-
error	执行上传请求出现异常的回调（一般为网络异常、URL 404等）。返回两个参数，分别为：*index*（当前文件的索引）、*upload*（重新上传的方法）。详见下文	function	-

1.25.3 上传接口

设定一个 URL 地址给 url 参数，用来告诉 upload 模块的服务端上传接口，示例代码如

图 1-xx 所示。

```
code
01. upload.render({
02.   elem: '#id'
03.   ,url: '/api/upload/' //必填项
04.   ,method: ''  //可选项。HTTP类型，默认post
05.   ,data: {} //可选项。额外的参数，如: {id: 123, abc: 'xxx'}
06. });
07.
```

该接口返回的相应信息（response）必须是一个标准的 JSON 格式，示例代码如图 1-xx 所示。

```
Response
01. {
02.   "code": 0
03.   ,"msg": ""
04.   ,"data": {
05.     "src": "http://cdn.layui.com/123.jpg"
06.   }
07. }
08.
```

注意 1：不一定非得按照上述格式返回，只要是合法的 JSON 字符即可。其响应信息会转化成 JS 对象传递给 done 回调。

注意 2：如果上传后，出现文件下载框（一般为 ie 下），那么需要在服务端对 response 的 header 设置 Content-Type: text/html。

1.25.4 实现基本上传功能

前端代码如下：

```
<!DOCTYPE html>
<html>
    <head>
```

```
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
        </button>
        <script>
            var upload = layui.upload;
            //执行实例
            var uploadInst = upload.render({
                elem: '#test1', //绑定元素
                url: 'Test1', //上传接口
                done: function (res) {
                    //上传完毕回调
                    alert("成功上传");
                },
                error: function () {
                    //请求异常回调
                    alert("上传失败");
                }
            });
        </script>
    </body>
</html>
```

DTO 类代码如下：

```
package dto;

public class UploadResponse {
    private int code;
    private String msg;
    private Object data;

    public UploadResponse() {
    }

    public UploadResponse(int code, String msg, Object data) {
        this.code = code;
        this.msg = msg;
        this.data = data;
```

```
    }

    public int getCode() {
        return code;
    }

    public void setCode(int code) {
        this.code = code;
    }

    public String getMsg() {
        return msg;
    }

    public void setMsg(String msg) {
        this.msg = msg;
    }

    public Object getData() {
        return data;
    }

    public void setData(Object data) {
        this.data = data;
    }
}
```

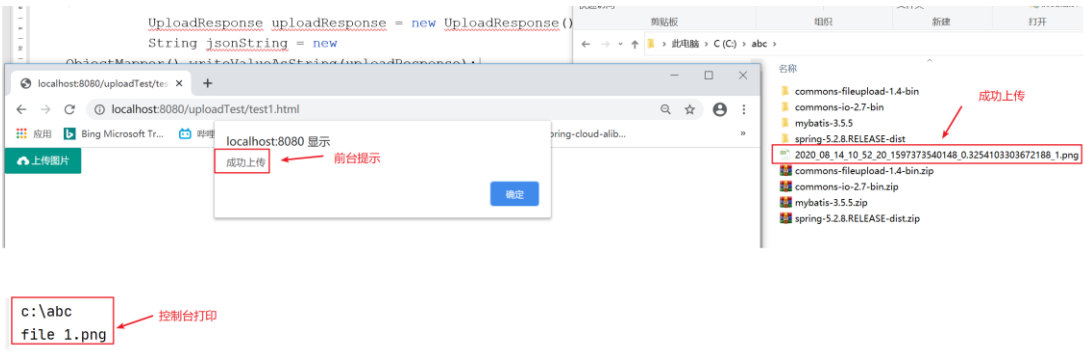
Servlet 代码如下：

```
package controller;

import com.fasterxml.jackson.databind.ObjectMapper;
import dto.UploadResponse;
import org.apache.commons.fileupload.FileItem;
import org.apache.commons.fileupload.FileUploadException;
import org.apache.commons.fileupload.disk.DiskFileItemFactory;
import org.apache.commons.fileupload.servlet.ServletFileUpload;

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.*;
```

```
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;

@WebServlet(name = "Test1", urlPatterns = "/Test1")
public class Test1 extends HttpServlet {
    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException
{
        try {
            String uploadPath = "c:\\abc";
            System.out.println(uploadPath);
            SimpleDateFormat format = new
SimpleDateFormat("yyyy_MM_dd_HH_mm_ss");

            request.setCharacterEncoding("utf-8");
            DiskFileItemFactory factory = new DiskFileItemFactory();
            ServletContext servletContext =
this.getServletConfig().getServletContext();
            ServletFileUpload upload = new ServletFileUpload(factory);
            List<FileItem> items = upload.parseRequest(request);

            for (int i = 0; i < items.size(); i++) {
                FileItem fileItem = items.get(i);
                if (fileItem.isFormField()) {
                    System.out.println(fileItem.getFieldName() + " "
                            + new
String(fileItem.getString().getBytes("iso-8859-1"), "utf-8"));
                } else {
                    if (fileItem.getInputStream().available() != 0) {
                        Date nowDate = new Date();
                        String uploadFileName = format.format(nowDate);
                        uploadFileName = uploadFileName + "_" +
System.currentTimeMillis() + "_" + Math.random() + "_"
                                + fileItem.getName();
                        System.out.println(fileItem.getFieldName() + "
" + fileItem.getName());

                        InputStream in = fileItem.getInputStream();
                        OutputStream out = new FileOutputStream(new
File(uploadPath, uploadFileName));
                        byte b[] = new byte[1024];
                        int len = -1;
                        while ((len = in.read(b)) != -1) {
```

```
                    out.write(b, 0, len);
                }
                out.close();
                in.close();
            }
        }
    }

    UploadResponse uploadResponse = new UploadResponse();
    String jsonString = new
ObjectMapper().writeValueAsString(uploadResponse);

    response.setContentType("text/html;charset=utf-8");
    PrintWriter out = response.getWriter();
    out.print(jsonString);
    out.flush();
    out.close();
} catch (FileUploadException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
}
}
```

运行效果如图 1-xx 所示。



1.25.5 属性 data 和 headers

前端代码如下：

```
<!DOCTYPE html>
<html>
    <head>
```

```
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
        </button>
        <script>
            var upload = layui.upload;
            //执行实例
            var uploadInst = upload.render({
                elem: '#test1', //绑定元素
                url: 'Test2', //上传接口
                data: {"username": "中国", "password": "中国人"},
                headers: {"token": 'gaohongyan_token_token'},
                done: function (res) {
                    //上传完毕回调
                    alert("成功上传");
                },
                error: function () {
                    //请求异常回调
                    alert("上传失败");
                }
            });
        </script>
    </body>
</html>
```

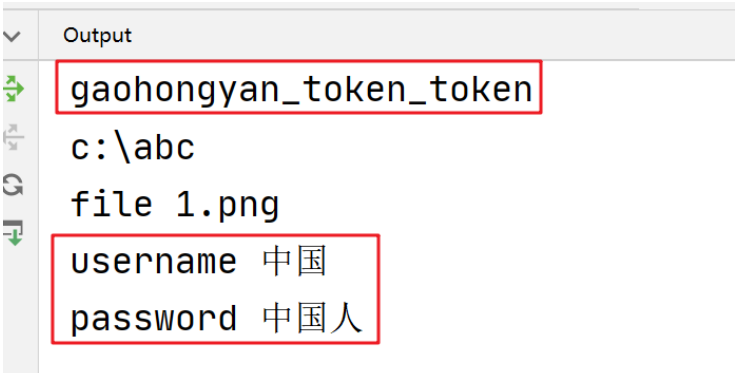
Servlet 代码如下：

```
package controller;

import com.fasterxml.jackson.databind.ObjectMapper;
import dto.UploadResponse;
import org.apache.commons.fileupload.FileItem;
import org.apache.commons.fileupload.FileUploadException;
import org.apache.commons.fileupload.disk.DiskFileItemFactory;
import org.apache.commons.fileupload.servlet.ServletFileUpload;

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
```

```
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.*;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;

@WebServlet(name = "Test2", urlPatterns = "/Test2")
public class Test2 extends HttpServlet {
    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException
{
        try {
            System.out.println(request.getHeader("token"));
            String uploadPath = "c:\\abc";
            System.out.println(uploadPath);
            SimpleDateFormat format = new
SimpleDateFormat("yyyy_MM_dd_HH_mm_ss");

            request.setCharacterEncoding("utf-8");
            DiskFileItemFactory factory = new DiskFileItemFactory();
            ServletContext servletContext =
this.getServletConfig().getServletContext();
            ServletFileUpload upload = new ServletFileUpload(factory);
            List<FileItem> items = upload.parseRequest(request);

            for (int i = 0; i < items.size(); i++) {
                FileItem fileItem = items.get(i);
                if (fileItem.isFormField()) {
                    System.out.println(fileItem.getFieldName() + " "
                            + new
String(fileItem.getString().getBytes("iso-8859-1"), "utf-8"));
                } else {
                    if (fileItem.getInputStream().available() != 0) {
                        Date nowDate = new Date();
                        String uploadFileName = format.format(nowDate);
                        uploadFileName = uploadFileName + "_" +
System.currentTimeMillis() + "_" + Math.random() + "_"
                                + fileItem.getName();
                        System.out.println(fileItem.getFieldName() + "
" + fileItem.getName());

                        InputStream in = fileItem.getInputStream();
```

```
                    OutputStream out = new FileOutputStream(new
File(uploadPath, uploadFileName));
                    byte b[] = new byte[1024];
                    int len = -1;
                    while ((len = in.read(b)) != -1) {
                        out.write(b, 0, len);
                    }
                    out.close();
                    in.close();
                }
            }
        }

        UploadResponse uploadResponse = new UploadResponse();
        String jsonString = new
ObjectMapper().writeValueAsString(uploadResponse);

        response.setContentType("text/html;charset=utf-8");
        PrintWriter out = response.getWriter();
        out.print(jsonString);
        out.flush();
        out.close();
    } catch (FileUploadException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
  }
}
```

运行效果如图 1-xx 所示。



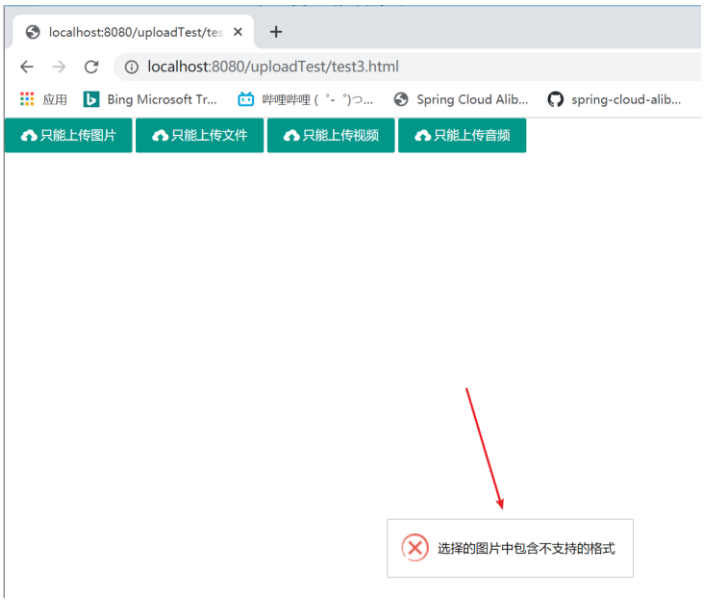
1.25.6 属性 accept

前端代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>只能上传图片
        </button>
        <button type="button" class="layui-btn" id="test2">
            <i class="layui-icon">&#xe67c;</i>只能上传文件
        </button>
        <button type="button" class="layui-btn" id="test3">
            <i class="layui-icon">&#xe67c;</i>只能上传视频
        </button>
        <button type="button" class="layui-btn" id="test4">
            <i class="layui-icon">&#xe67c;</i>只能上传音频
        </button>
        <script>
            //accept 可选值为：images（图片）、file（所有文件）、video（视频）、
audio（音频）
            var upload = layui.upload;
            var uploadInst1 = upload.render({
                elem: '#test1',
                url: 'AnyPath',
                accept: 'images',
            });
            var uploadInst2 = upload.render({
                elem: '#test2',
                url: 'AnyPath',
                accept: 'file',
            });
            var uploadInst3 = upload.render({
                elem: '#test3',
                url: 'AnyPath',
                accept: "video"
            });
```

```
            var uploadInst4 = upload.render({
                elem: '#test4',
                url: 'AnyPath',
                accept: "audio"
            });
        </script>
    </body>
</html>
```

上传的文件类型没有通过校验，如图 1-xx 所示。



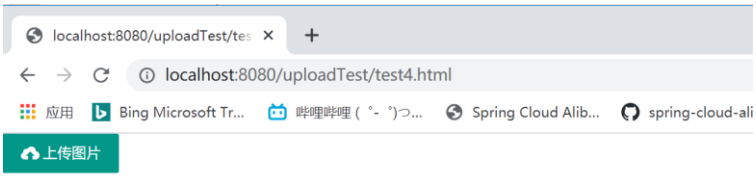
1.25.7 属性 exts

前端代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>只能上传 png 图片
        </button>
        <script>
            var upload = layui.upload;
```

```
            var uploadInst1 = upload.render({
                elem: '#test1',
                url: 'AnyPath',
                accept: 'images',
                exts: 'png'
            });
        </script>
    </body>
</html>
```

上传非 png 格式出现异常，如图 1-xx 所示。



也支持多个扩展名：

exts: 'zip|rar|pdf|jpg|doc|docx|xls|xlsx'

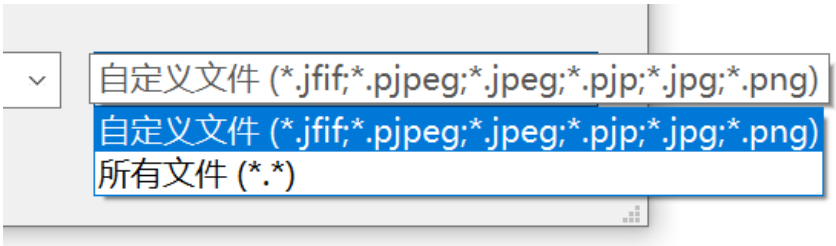
1.25.8 属性 acceptMime

前端代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
```

```
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
        </button>
        <script>
            var upload = layui.upload;
            var uploadInst1 = upload.render({
                elem: '#test1',
                url: 'AnyPath',
                acceptMime: "image/jpeg,image/png"//只能上传 jpeg 或 png
格式
            });
        </script>
    </body>
</html>
```

Chrome 浏览器运行效果如图 1-xx 所示。



可以过滤文件类型，只显示 jpeg 或 png 格式的文件。

1.25.9 属性 auto 和 bindAction

前端代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
        </button>
        <button type="button" class="layui-btn" id="beginUpload">
            开始上传
```

```
        </button>
        <script>
            var upload = layui.upload;
            var uploadInst1 = upload.render({
                elem: '#test1',
                url: 'AnyPath',
                auto: false,
                bindAction: "#beginUpload"
            });
        </script>
    </body>
</html>
```

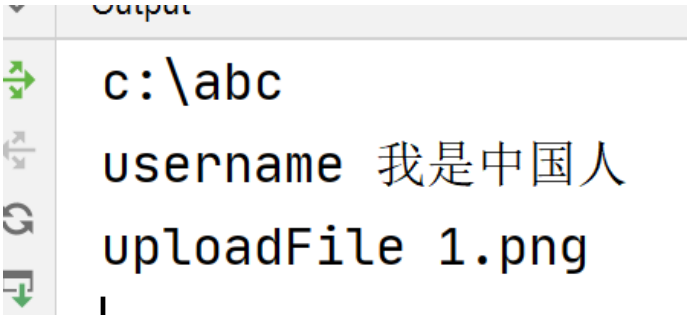
1.25.10 结合其它表单处理上传

前端代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-form-item">
            <label class="layui-form-label">输入框</label>
            <div class="layui-input-block">
                <input type="text" id="username" placeholder="请输入账
号" autocomplete="off" class="layui-input">
            </div>
        </div>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
        </button>
        <button type="button" class="layui-btn" id="beginUpload">
            开始上传
        </button>
        <script>
            var upload = layui.upload;
            var uploadInst1 = upload.render({
                elem: '#test1',
                auto: false,
```

```
            });
            $(document).ready(function () {
                $("#beginUpload").click(function () {
                    var usernameValue = $("#username").val();
                    var inputFile = $("input[type=file]")[0].files[0];

                    var formData = new FormData();
                    formData.append('username', usernameValue)
                    formData.append('uploadFile', inputFile);

                    $.ajax({
                        url: 'Test1?t=' + new Date().getTime(),
                        type: 'POST',
                        data: formData,
                        processData: false,
                        contentType: false,
                        success: function () {
                            alert("上传成功！");
                        }
                    });
                });
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



此案例是 ajax 无刷新版上传，使用<form></form>并更改相应的代码可以实现有刷新上传。

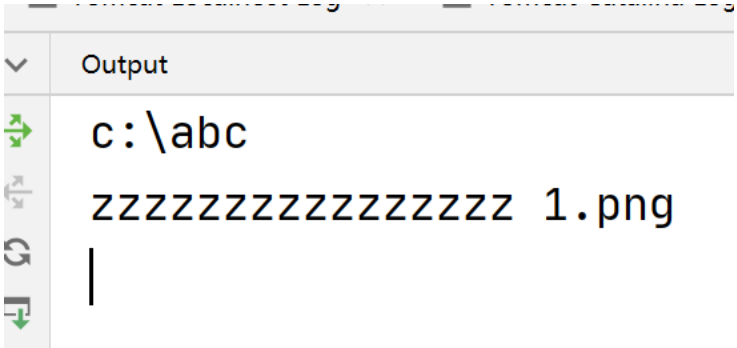
1.25.11 属性 field

前端代码如下：

```
<!DOCTYPE html>
<html>
```

```
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
        </button>
        <script>
            //accept 可选值为：images（图片）、file（所有文件）、video（视频）、
audio（音频）
            var upload = layui.upload;
            var uploadInst1 = upload.render({
                elem: '#test1',
                url: 'Test1',
                field: "zzzzzzzzzzzzzzz"
            });
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



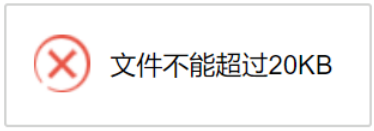
1.25.12 属性 size

前端代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
```

```
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
        </button>
        <script>
            var upload = layui.upload;
            var uploadInst1 = upload.render({
                elem: '#test1',
                url: 'Test1',
                size: 20
            });
        </script>
    </body>
</html>
```

运行结果如图 1-xx 所示。



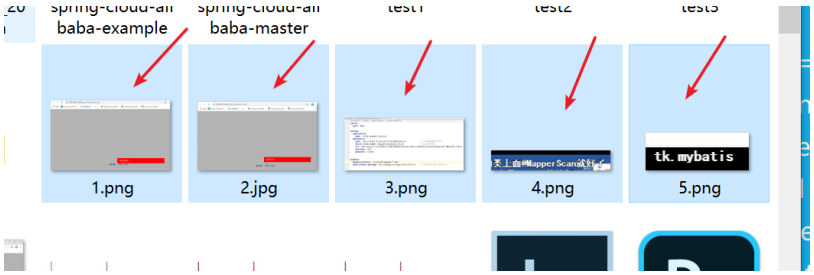
1.25.13 属性 multiple

前端代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
        </button>
        <script>
            var upload = layui.upload;
            var uploadInst1 = upload.render({
```

```
                elem: '#test1',
                url: 'Test1',
                auto: false,
                multiple: true
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



同时上传 5 个文件。

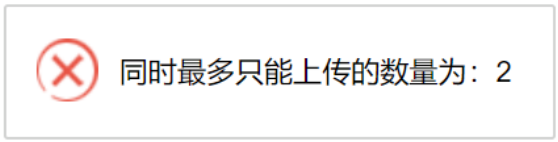
1.25.14 属性 number

前端代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
        </button>
        <script>
            var upload = layui.upload;
            var uploadInst1 = upload.render({
                elem: '#test1',
                url: 'Test1',
                auto: false,
                multiple: true,
                number: 2
```

```
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.25.15 属性 drag

upload 组件默认支持拖拽，本案例进行了禁用。

前端代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
        </button>
        <script>
            var upload = layui.upload;
            var uploadInst1 = upload.render({
                elem: '#test1',
                url: 'Test1',
                multiple: true,
                drag: false
            });
        </script>
    </body>
</html>
```

当把图片拖拽到上传组件时，并不是上传，而是在浏览器中打开图片，因为拖拽被禁用了。

1.25.16 回调 choose

在文件被选择后触发，该回调会在 before 回调之前执行。一般用于非自动上传（即 auto:false）的场景，比如预览图片等。

示例代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
        </button>
        <script>
            var upload = layui.upload;
            var uploadInst1 = upload.render({
                elem: '#test1',
                auto: false,
                multiple: true,
                choose: function (obj) {
                    //将每次选择的文件追加到文件队列
                    var files = obj.pushFile();
                    //预读本地文件，如果是多文件，则会遍历。(不支持 ie8/9)
                    obj.preview(function (index, file, result) {
                        console.log(index); //得到文件索引
                        console.log(file); //得到文件对象
                        console.log(result); //得到文件 base64 编码，比如图
片

                        //obj.resetFile(index, file, '123.jpg'); //重命
名文件名

                        //这里还可以做一些 append 文件列表 DOM 的操作
                        //obj.upload(index, file); //对上传失败的单个文件
重新上传，一般在某个事件中使用
                        //delete files[index]; //删除列表中对应的文件，一般
在某个事件中使用
                    });
                }
```

```
            });
        </script>
    </body>
</html>
```

事实上这是一个非常实用的功能，可轻松应对复杂的列表文件上传管理，示例代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-upload">
            <button type="button" class="layui-btn layui-btn-normal"
id="testList">选择多文件</button>
            <div class="layui-upload-list">
                <table class="layui-table">
                    <thead>
                        <th>文件名</th>
                        <th>大小</th>
                        <th>状态</th>
                        <th>操作</th>
                    </thead>
                    <tbody id="demoList"></tbody>
                </table>
            </div>
            <button type="button" class="layui-btn"
id="testListAction">开始上传</button>
        </div>
        <script>
            //多文件列表示例
            var demoListView = $('#demoList');
            var upload = layui.upload;
            var uploadListIns = upload.render({
                elem: '#testList',
                url: 'Test1',
                accept: 'file',
                multiple: true,
                auto: false,
                bindAction: '#testListAction',
```

```
            choose: function (obj) {
                var files = this.files = obj.pushFile(); //将每次选
择的文件追加到文件队列
                //读取本地文件
                obj.preview(function (index, file, result) {
                    var tr = $(['<tr id="upload-' + index + '">'
                        , '<td>' + file.name + '</td>'
                        , '<td>' + (file.size / 1024).toFixed(1) +
'kb</td>'
                        , '<td>等待上传</td>'
                        , '<td>'
                        , '<button class="layui-btn layui-btn-xs
demo-reload layui-hide">重传</button>'
                        , '<button class="layui-btn layui-btn-xs
layui-btn-danger demo-delete">删除</button>'
                        , '</td>'
                        , '</tr>'].join(''));

                    //单个重传
                    tr.find('.demo-reload').on('click', function ()
{
                        obj.upload(index, file);
                    });

                    //删除
                    tr.find('.demo-delete').on('click', function ()
{
                        delete files[index]; //删除对应的文件
                        tr.remove();
                        uploadListIns.config.elem.next()[0].value =
''; //清空 input file 值，以免删除后出现同名文件不可选
                    });

                    demoListView.append(tr);
                });
            },
            done: function (res, index, upload) {
                if (res.code == 0) { //上传成功
                    var tr = demoListView.find('tr#upload-' + index)
                        , tds = tr.children();
                    tds.eq(2).html('<span style="color: #5FB878;">
上传成功</span>');
                    tds.eq(3).html(''); //清空操作
                    return delete this.files[index]; //删除文件队列已
```

```
经上传成功的文件
                }
                this.error(index, upload);
            },
            error: function (index, upload) {
                var tr = demoListView.find('tr#upload-' + index)
                    , tds = tr.children();
                tds.eq(2).html('<span style="color: #FF5722;">上传
失败</span>');

tds.eq(3).find('.demo-reload').removeClass('layui-hide'); //显示重
传
            }
        });
    </script>
</body>
</html>
```

1.25.17 回调 before

在 choose 回调之后、done/error 回调之前触发。返回的参数完全类似 choose 回调。一般用于上传完毕前的 loading、图片预览等。

前端代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
        </button>
        <script>
            var upload = layui.upload;
            //执行实例
            var uploadInst = upload.render({
                elem: '#test1', //绑定元素
                url: 'Test1', //上传接口
```

```
                before: function (obj) { //obj 参数包含的信息，跟 choose
回调完全一致，可参见上文。
                    layer.load(); //上传 loading
                },
                done: function (res, index, upload) {
                    layer.closeAll('loading'); //关闭 loading
                },
                error: function (index, upload) {
                    layer.closeAll('loading'); //关闭 loading
                }
            });
        </script>
    </body>
</html>
```

1.25.18 回调 done

在上传接口请求完毕后触发，但文件不一定是上传成功的，只是接口的响应状态正常（200）。回调返回三个参数，分别为：服务端响应信息、当前文件的索引、重新上传的方法。

前端代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
        </button>
        <script>
            var upload = layui.upload;
            //执行实例
            var uploadInst = upload.render({
                elem: '#test1', //绑定元素
                url: 'Test1', //上传接口
                multiple: true,
                before: function (obj) { //obj 参数包含的信息，跟 choose
```

```
回调完全一致，可参见上文。
                layer.load(); //上传 loading
            },
            done: function (res, index, upload) {
                alert('done run !');
                //假设 code=0 代表上传成功
                if (res.code == 0) {
                    //do something （比如将 res 返回的图片链接保存到表单的隐藏域）
                }

                //获取当前触发上传的元素，一般用于 elem 绑定 class 的情况，注意：此乃 layui 2.1.0 新增
                var item = this.item;

                //文件保存失败
                //do something

                var test = "123";
            }
        });
    </script>
</body>
</html>
```

1.25.19 回调 error

当请求上传时出现异常时触发（如网络异常、404/500 等）。回调返回两个参数，分别为：当前文件的索引、重新上传的方法。

前端代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
```

```
        </button>
        <div id="demoText"></div>
        <script>
            var upload = layui.upload;
            //执行实例
            var uploadInst = upload.render({
                elem: '#test1', //绑定元素
                url: 'TestXXXXXXXXXXXX', //错误的上传路径
                multiple: true,
                error: function (index, upload) {
                    //当上传失败时，可以生成一个"重新上传"的按钮，点击该按钮时，
执行 upload()方法即可实现重新上传。
                    //演示失败状态，并实现重传
                    var demoText = $('#demoText');
                    demoText.html('<span style="color: #FF5722;">上传失
败</span> <a class="layui-btn layui-btn-mini demo-reload">重试</a>');
                    demoText.find('.demo-reload').on('click',
function () {
                        uploadInst.upload();
                    });
                }
            });
        </script>
    </body>
</html>
```

如下写法效果一样:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
        </button>
        <div id="demoText"></div>
        <script>
            var upload = layui.upload;
            //执行实例
```

```
            upload.render({
                elem: '#test1', //绑定元素
                url: 'TestXXXXXXXXXXXX', //错误的上传路径
                multiple: true,
                error: function (index, upload) {
                    //当上传失败时，可以生成一个"重新上传"的按钮，点击该按钮时，
执行 upload()方法即可实现重新上传。
                    //演示失败状态，并实现重传
                    var demoText = $('#demoText');
                    demoText.html('<span style="color: #FF5722;">上传失
败</span> <a class="layui-btn layui-btn-mini demo-reload">重试</a>');
                    demoText.find('.demo-reload').on('click',
function () {
                        upload();
                    });
                }
            });
        </script>
    </body>
</html>
```

1.25.20 多文件上传完毕后的状态回调

只有当开启多文件时（即 multiple: true），该回调才会被触发。回调返回一个 object 类型的参数，包含一些状态数据。

示例代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
        </button>
        <script>
            var upload = layui.upload;
            var uploadInst1 = upload.render({
                elem: '#test1',
```

```
            url: 'Test1',
            multiple: true,
            allDone: function (obj) { //当文件全部被提交后，才触发
                console.log(obj.total); //得到总文件数
                console.log(obj.successful); //请求成功的文件数
                console.log(obj.aborted); //请求失败的文件数
            },
            done: function (res, index, upload) {
                //每个文件提交一次触发一次
                console.log("done " + index);
            }
        });
    </script>
</body>
</html>
```

执行效果如图 1-xx 所示。

```
done 1597385457016-0
done 1597385457016-1
2
2
0
```

1.25.21 文件上传进度的回调

在网速一般的情况下，大文件的上传通常需要一定时间的等待，而浏览器并不会醒目地告知它正在努力地上传中，此时为了提升用户体验，可以通过该回调制作一个进度条。

示例代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
        </button>
```

```
        <div class="layui-progress" lay-filter="demo">
            <div class="layui-progress-bar"></div>
        </div>
        <script>
            var upload = layui.upload;
            var element = layui.element;
            upload.render({
                elem: '#test1',
                url: 'Test1',
                multiple: true,
                progress: function (n, elem) {
                    var percent = n + '%' //获取进度百分比
                    element.progress('demo', percent); //可配合 layui 进度条元素使用
                    console.log(elem); //得到当前触发的元素 DOM 对象。可通过该元素定义的属性值匹配到对应的进度条。
                }
            });
        </script>
    </body>
</html>
```

1.25.22 重载实例

有时可能需要对 upload.render()进行重载，通过改变一些参数（如将上传文件重置为只上传图片等场景）来重置功能，示例代码如图 1-xx 所示。

```
code
01. //创建一个实例
02. var uploadInst = upload.render({
03.   elem: '#id'
04.   ,url: '/api/upload/'
05.   ,size: 1024*5 //限定大小
06. });
07.
08. //重载该实例，支持重载全部基础参数
09. uploadInst.reload({
10.   accept: 'images' //只允许上传图片
11.   ,acceptMime: 'image/*' //只筛选图片
12.   ,size: 1024*2 //限定大小
13. });
14.
```

注意：该方法为 layui 2.5.0 开始新增

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css"/>
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <button type="button" class="layui-btn" id="test1">
            <i class="layui-icon">&#xe67c;</i>上传图片
        </button>
        <input type="button" id="uploadJPEG" value="只能上传 jpeg 文件
"/>
        <script>
            var upload = layui.upload;
            var element = layui.element;
            var myupload = upload.render({
                elem: '#test1',
                url: 'Test1',
                multiple: true,
```

```
                accept: 'images',
                exts: 'png',
                progress: function (n, elem) {
                    var percent = n + '%' //获取进度百分比
                    element.progress('demo', percent); //可配合 layui 进度条元素使用
                    console.log(elem); //得到当前触发的元素 DOM 对象。可通过该元素定义的属性值匹配到对应的进度条。
                }
            });
            $(document).ready(function () {
                $("#uploadJPEG").click(function () {
                    myupload.reload({
                        multiple: true,
                        accept: 'images',
                        exts: 'jpg|jpeg'
                    });
                });
            });
        </script>
    </body>
</html>
```

点击 button 后，upload 组件由原来的只能上传 png 改成只能上传 jpg 文件。

1.25.23 重新上传

在执行 upload.render(options)方法时，其实有返回一个实例对象，以便实现重新上传等操作。注意：这是对当前上传队列的全局重新上传，而 choose 回调返回的 obj.upload(index, file)方法则是对单个文件进行重新上传，示例代码如图 1-xx 所示。

```
code
var uploadInst = upload.render({
  elem: '#id'
  ,url: '/api/upload/'
  ,choose: function(obj){
    obj.preview(function(index, file, result){
      //对上传失败的单个文件重新上传，一般在某个事件中使用
      //obj.upload(index, file);
    });
  }
});

//重新上传的方法，一般在某个事件中使用
//uploadInst.upload();

```

手动重新上传的案例前面章节已经进行了介绍。

1.26 穿梭框组件 transfer

模块加载名称：transfer。

1.26.1 快速使用

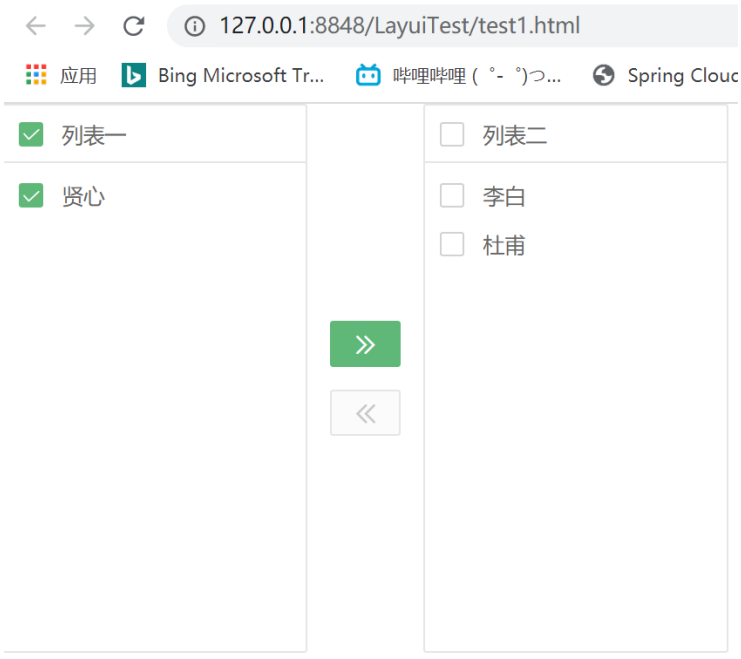
transfer 组件可以进行数据的交互筛选。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var transfer = layui.transfer;
            //渲染
            transfer.render({
                elem: '#test1' //绑定元素
                    ,
```

```
            data: [{
                "value": "1",
                "title": "李白",
                "disabled": "",
                "checked": ""
            }, {
                "value": "2",
                "title": "杜甫",
                "disabled": "",
                "checked": ""
            }, {
                "value": "3",
                "title": "贤心",
                "disabled": "",
                "checked": ""
            }],
            id: 'demo1' //定义索引
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.26.2 基础参数

目前 transfer 组件提供如图 1-xx 所示的基础参数，可根据需要进行相应的设置。

参数选项	说明	类型	默认值
elem	指向容器选择器	String/Object	-
title	穿梭框上方标题	Array	['标题一', '标题二']
data	数据源	Array	[{}, {}, ...]
parseData	用于对数据源进行格式解析	Function	详见数据源格式解析
value	初始选中的数据（右侧列表）	Array	-
id	设定实例唯一索引，用于基础方法传参使用。	String	-
showSearch	是否开启搜索	Boolean	false
width	定义左右穿梭框宽度	Number	200
height	定义左右穿梭框高度	Number	340
text	自定义文本，如空数据时的异常提示等。 code layui.code 01. text: { 02. none: '无数据' //没有数据时的文案 03. ,searchNone: '无匹配数据' //搜索无匹配数据时的文案 04. } 05.	Object	-
onchange	左右数据穿梭时的回调	Function	详见穿梭时的回调

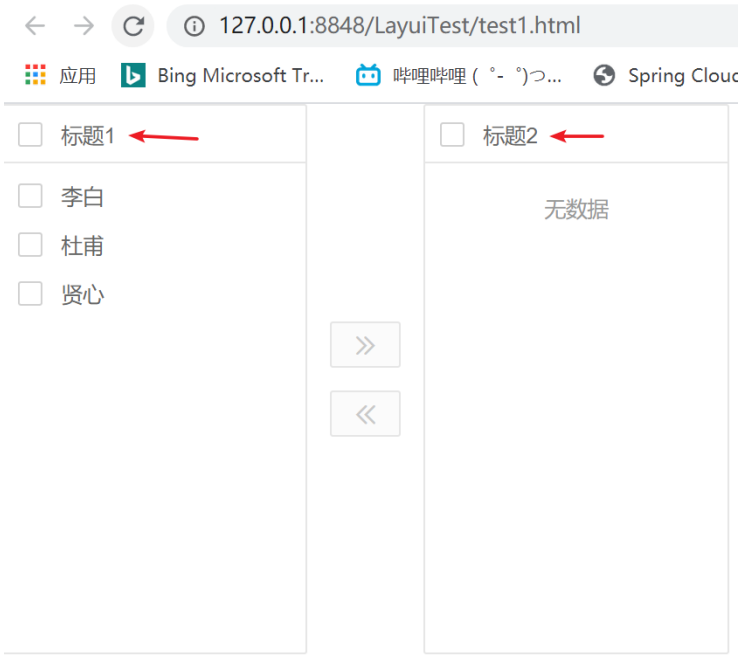
1.26.3 属性 title

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var transfer = layui.transfer;
            //渲染
            transfer.render({
                elem: '#test1', //绑定元素
                title: ['标题1', '标题2'],
                data: [{
                    "value": "1",
                    "title": "李白",
```

```
                "disabled": "",
                "checked": ""
            }, {
                "value": "2",
                "title": "杜甫",
                "disabled": "",
                "checked": ""
            }, {
                "value": "3",
                "title": "贤心",
                "disabled": "",
                "checked": ""
            }],
            id: 'demo1' //定义索引
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



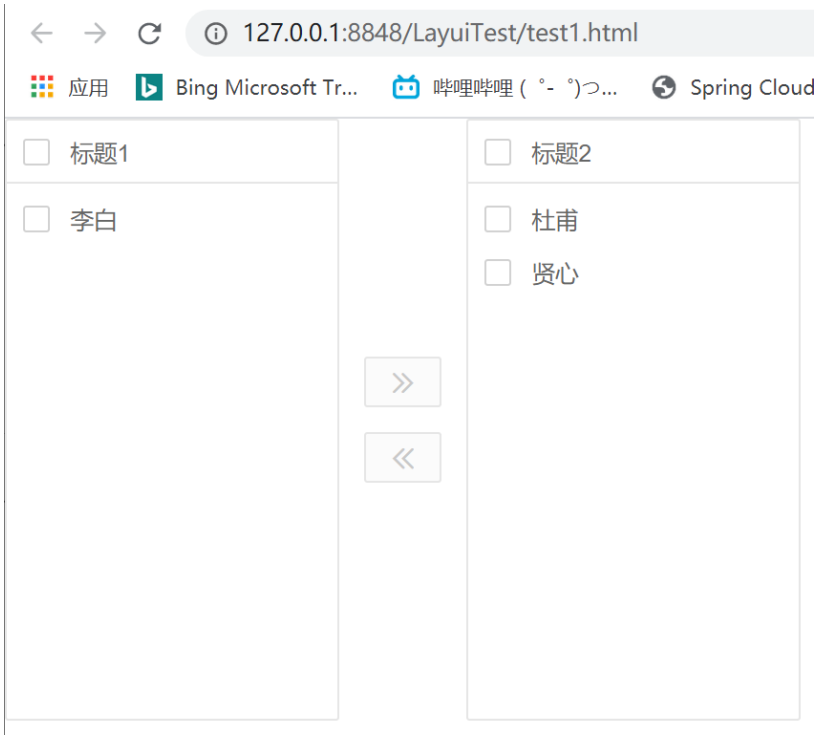
1.26.4 属性 value

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
```

```
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var transfer = layui.transfer;
            //渲染
            transfer.render({
                elem: '#test1', //绑定元素
                title: ['标题1', '标题2'],
                data: [{
                    "value": "1",
                    "title": "李白",
                    "disabled": "",
                    "checked": ""
                }, {
                    "value": "2",
                    "title": "杜甫",
                    "disabled": "",
                    "checked": ""
                }, {
                    "value": "3",
                    "title": "贤心",
                    "disabled": "",
                    "checked": ""
                }],
                value: [2, 3],
                id: 'demo1' //定义索引
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



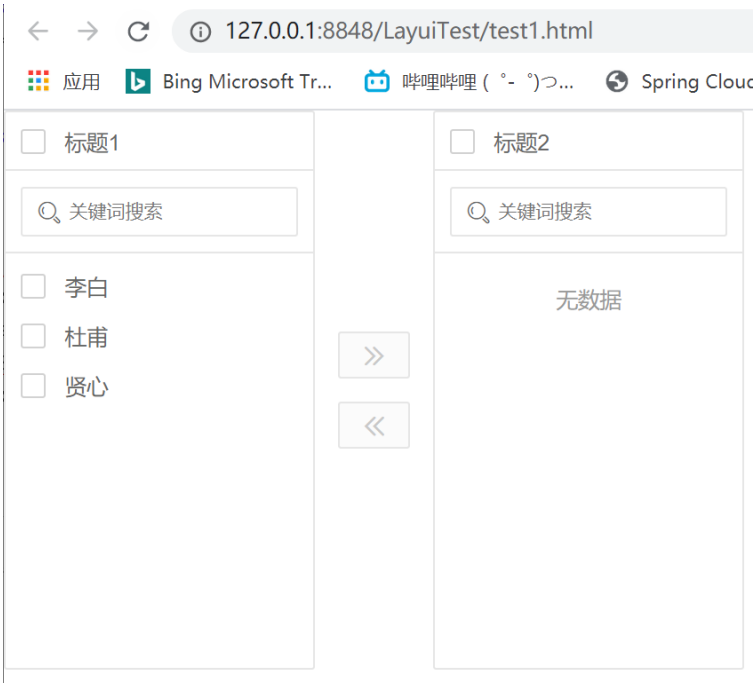
1.26.5 属性 showSearch

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var transfer = layui.transfer;
            //渲染
            transfer.render({
                elem: '#test1', //绑定元素
                title: ['标题1', '标题2'],
                showSearch: true,
                data: [{
                    "value": "1",
                    "title": "李白",
                    "disabled": "",
```

```
                "checked": ""
            }, {
                "value": "2",
                "title": "杜甫",
                "disabled": "",
                "checked": ""
            }, {
                "value": "3",
                "title": "贤心",
                "disabled": "",
                "checked": ""
            }],
            id: 'demo1' //定义索引
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



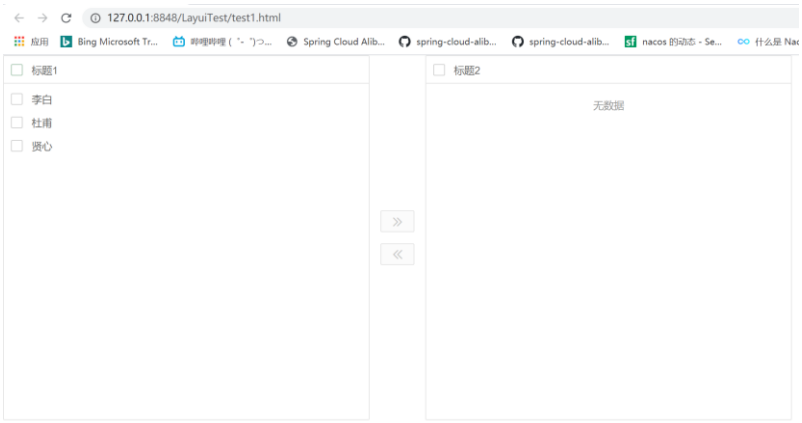
1.26.6 属性 width 和 height

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
```

```
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var transfer = layui.transfer;
            //渲染
            transfer.render({
                elem: '#test1', //绑定元素
                title: ['标题1', '标题2'],
                data: [{
                    "value": "1",
                    "title": "李白",
                    "disabled": "",
                    "checked": ""
                }, {
                    "value": "2",
                    "title": "杜甫",
                    "disabled": "",
                    "checked": ""
                }, {
                    "value": "3",
                    "title": "贤心",
                    "disabled": "",
                    "checked": ""
                }],
                width: 500,
                height: 500,
                id: 'demo1' //定义索引
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

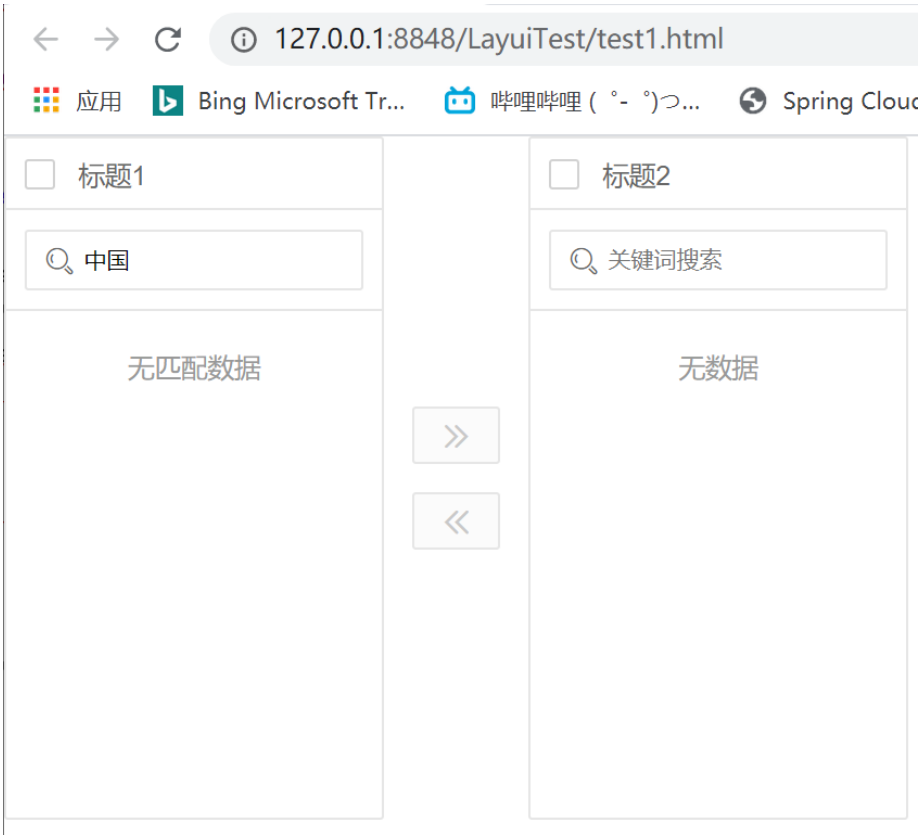


1.26.7 属性 text

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var transfer = layui.transfer;
            //渲染
            transfer.render({
                elem: '#test1', //绑定元素
                title: ['标题1', '标题2'],
                data: [],
                showSearch: true,
                text: {
                    none: '无数据', //没有数据时的文案
                    searchNone: '无匹配数据' //搜索无匹配数据时的文案
                },
                id: 'demo1' //定义索引
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.26.8 属性 parseData

数据格式解析的回调函数，用于将任意数据格式解析成 transfer 组件规定的数据格式，如图 1-xx 所示的是合法的数据格式。

合法的数据格式

```
[
  {"value": "1", "title": "李白", "disabled": "", "checked": ""}
  ,{"value": "2", "title": "杜甫", "disabled": "", "checked": ""}
  ,{"value": "3", "title": "贤心", "disabled": "", "checked": ""}
]

```

然而很多时候返回的数据格式可能并不符合规范，如图 1-xx 所示。

不符合规范的数据格式

```
[
  {"id": "1", "name": "李白"}
  ,{"id": "2", "name": "杜甫"}
  ,{"id": "3", "name": "贤心"}
]
```

这时候需要使用属性 parseData 将其解析成 transfer 组件所规定的数据格式。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var transfer = layui.transfer;
            //渲染
            transfer.render({
                elem: '#test1', //绑定元素
                title: ['标题1', '标题2'],
                data: [{
                    "id": "1",
                    "name": "李白"
                }, {
                    "id": "2",
                    "name": "杜甫"
                }, {
                    "id": "3",
                    "name": "贤心"
                }],
                parseData: function(res) {
```

```
                return {
                    "value": res.id, //数据值
                    "title": res.name, //数据标题
                    "disabled": res.disabled, //是否禁用
                    "checked": res.checked //是否选中
                }
            },
            id: 'demo1' //定义索引
        });
    </script>
</body>
</html>
```

1.26.9 左右穿梭的回调

当数据在左右穿梭时触发，回调返回当前被穿梭的数据。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var transfer = layui.transfer;
            //渲染
            transfer.render({
                elem: '#test1', //绑定元素
                title: ['标题1', '标题2'],
                data: [{
                    "value": "1",
                    "title": "李白",
                    "disabled": "",
                    "checked": ""
                }, {
                    "value": "2",
                    "title": "杜甫",
```

```
                "disabled": "",
                "checked": ""
            }, {
                "value": "3",
                "title": "贤心",
                "disabled": "",
                "checked": ""
            }],
            onchange: function(data, index) {
                console.log(data); //得到当前被穿梭的数据
                console.log(index); //如果数据来自左边，index为0，否则为
s1
            },
            id: 'demo1' //定义索引
        });
    </script>
</body>
</html>
```

1.26.10 基础方法

目前所开放的所有基础方法如图 1-xx 所示。

```
code
var transfer = layui.transfer;

transfer.set(options); //设定全局默认参数。options 即各项基础参数
transfer.getData(id); //获得右侧数据
transfer.reload(id, options); //重载实例

```

1.26.11 获得右侧数据

穿梭框的右侧数据通常被认为是选中数据，因此需要得到它并提交到后台。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
```

```
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <input type="button" id="mybutton" value="取值" />
        <script>
            var transfer = layui.transfer;
            //渲染
            transfer.render({
                elem: '#test1', //绑定元素
                title: ['标题1', '标题2'],
                data: [{
                    "value": "1",
                    "title": "李白",
                    "disabled": "",
                    "checked": ""
                }, {
                    "value": "2",
                    "title": "杜甫",
                    "disabled": "",
                    "checked": ""
                }, {
                    "value": "3",
                    "title": "贤心",
                    "disabled": "",
                    "checked": ""
                }],
                id: 'demo1' //定义索引
            });

            $(document).ready(function() {
                $("#mybutton").click(function() {
                    var getData = transfer.getData('demo1');
                    var test = "test";
                });

            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.26.12 重载

重载一个已经创建的组件实例，被覆盖新的基础属性。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <input type="button" id="mybutton" value="重载" />
        <script>
            var transfer = layui.transfer;
            //渲染
            transfer.render({
                elem: '#test1', //绑定元素
                title: ['标题1', '标题2'],
                data: [{
                    "value": "1",
                    "title": "李白",
                    "disabled": "",
                    "checked": ""
                }, {
                    "value": "2",
                    "title": "杜甫",
                    "disabled": "",
                    "checked": ""
                }, {
                    "value": "3",
                    "title": "贤心",
```

```
                "disabled": "",
                "checked": ""
            }],
            id: 'demo1' //定义索引
        });

        $(document).ready(function() {
            $("#mybutton").click(function() {
                //可以重载所有基础参数
                transfer.reload('demo1', {
                    title: ['新列表1', '新列表2']
                });
            });
        });
    </script>
</body>
</html>
```

1.27 颜色选择器 colorpicker

在主题定制的应用场景中，自然离不开颜色的自定义，而往往需要的是关于它的直观选择，colorpicker 支持 hex、rgb、rgba 三类色彩模式，在代码中简单的调用后，便可在网页系统中自由拖拽去选择中意的颜色。

模块加载名称：colorpicker。

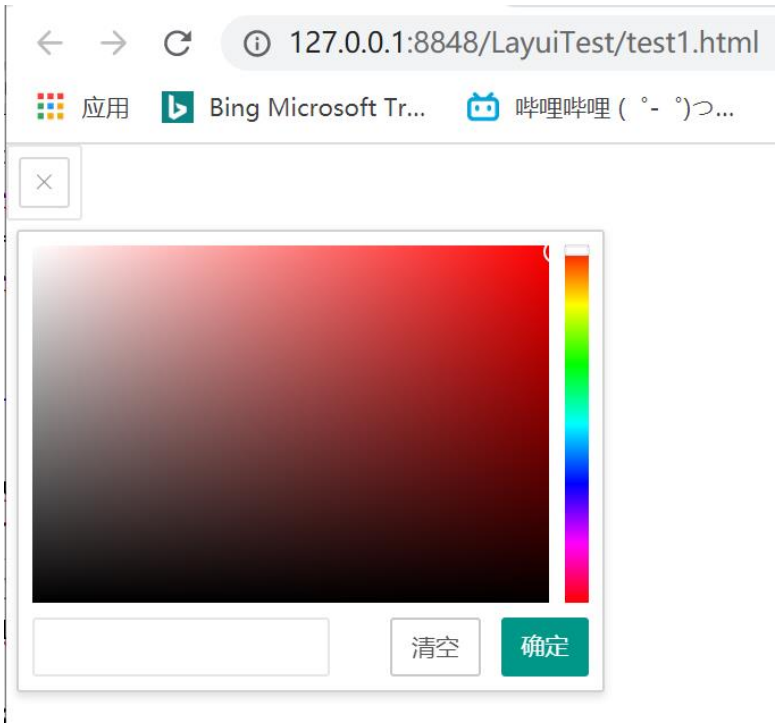
1.27.1 快速使用

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            layui.use('colorpicker', function() {
                var colorpicker = layui.colorpicker;
```

```
            //渲染
            colorpicker.render({
                elem: '#test1' //绑定元素
            });
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



1.27.2 基础参数

colorpicker 组件目前支持如图 1-xx 所示的参数。

参数选项	说明	类型	默认值
elem	指向容器选择器	string/object	-
color	默认颜色，不管你是使用 hex、rgb 还是 rgba 的格式输入，最终会以指定的格式显示。	string	-
format	颜色显示/输入格式，可选值： *hex* 、 *rgb* 若在 rgb 格式下开启了透明度，格式会自动变成 rgba。在没有输入颜色的前提下，组件会默认为 #000 也就是黑色。	string	hex（即 16 进制色值）
alpha	是否开启透明度，若不开启，则不会显示透明框。开启了透明度选项时，当你的默认颜色为 hex 或 rgb 格式，组件会默认加上值为 1 的透明度。相同的，当你没有开启透明度，却以 rgba 格式设置默认颜色时，组件会默认没有透明度。 注意： *该参数必须配合 rgba 颜色值使用*	boolean	false
predefine	预定义颜色是否开启	boolean	false
colors	预定义颜色，此参数需配合 *predefine: true* 使用。	Array	此处列举一部分：['#ff4500','#1e90ff','rgba(255, 69, 0, 0.68)','rgb(255, 120, 0)']
size	下拉框大小，可以选择：lg、sm、xs。	string	-

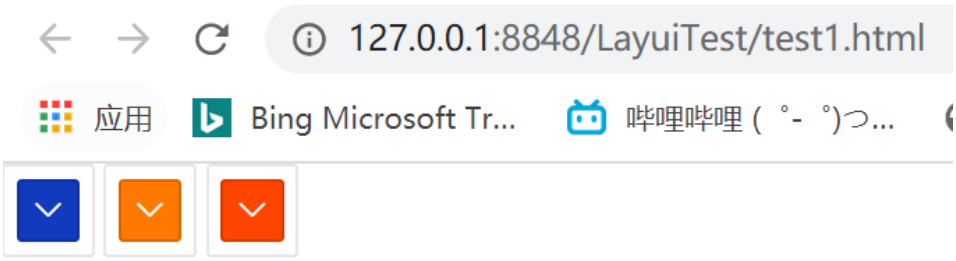
1.27.3 属性 color

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <div id="test2"></div>
        <div id="test3"></div>
        <script>
            var colorpicker = layui.colorpicker;
            //渲染
            colorpicker.render({
                elem: '#test1',
                color: "#123ABC"
            });
            colorpicker.render({
                elem: '#test2',
                color: "rgb(255, 120, 0)"
```

```
            });
            colorpicker.render({
                elem: '#test3',
                color: "rgba(255, 69, 0, 0.68)"
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



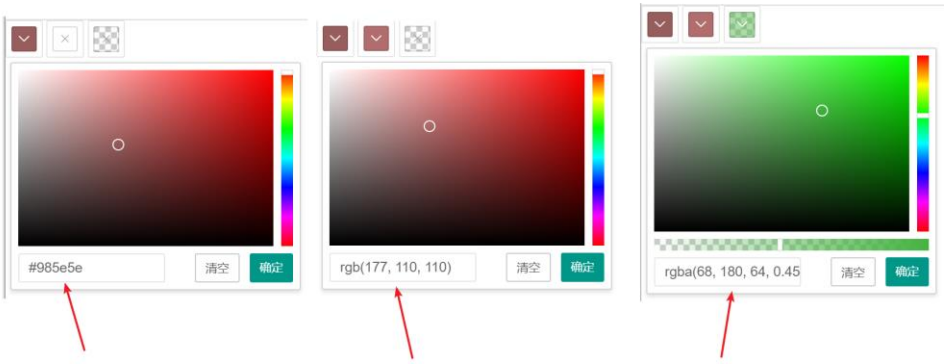
1.27.4 属性 format 和 alpha

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <div id="test2"></div>
        <div id="test3"></div>
        <script>
            var colorpicker = layui.colorpicker;
            //渲染
            colorpicker.render({
                elem: '#test1',
                format: "hex"
            });
            colorpicker.render({
                elem: '#test2',
```

```
            format: "rgb"
        });
        colorpicker.render({
            elem: '#test3',
            alpha: true,
            format: "rgb"
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



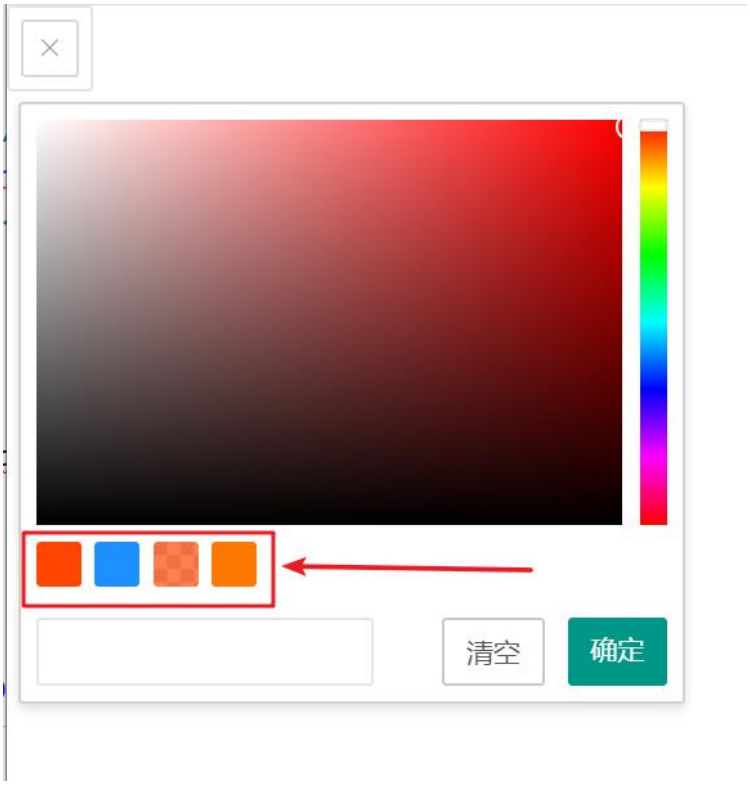
1.27.5 属性 predefine 和 colors

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var colorpicker = layui.colorpicker;
            //渲染
            colorpicker.render({
                elem: '#test1',
                predefine: true,
                colors: ['#ff4500', '#1e90ff', 'rgba(255, 69, 0, 0.68)',
```

```
'rgb(255, 120, 0)']
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.27.6 属性 size

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <div id="test2"></div>
        <div id="test3"></div>
```

```
        <script>
            var colorpicker = layui.colorpicker;
            //渲染
            colorpicker.render({
                elem: '#test1',
                size: "lg"
            });
            colorpicker.render({
                elem: '#test2',
                size: "sm"
            });
            colorpicker.render({
                elem: '#test3',
                size: "xs"
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

1.27.7 颜色被改变的回调

回调名：change。

当颜色在选择器中发生选择改变时，会进入 change 回调，可以通过它来进行所需的操作，如下代码就是实时的输出当前选择器的颜色。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
```

```
            var colorpicker = layui.colorpicker;
            //渲染
            colorpicker.render({
                elem: '#test1',
                change: function(color) {
                    console.log(color)
                }
            });
        </script>
    </body>
</html>
```

1.27.8 颜色选择后的回调

回调名：done。

点击颜色选择器的“确认”和“清除”按钮，均会触发 done 回调，回调返回当前选择的颜色值。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var colorpicker = layui.colorpicker;
            //渲染
            colorpicker.render({
                elem: '#test1',
                done: function(color) {
                    console.log(color)
                    //譬如你可以在回调中把得到的color赋值给表单
                }
            });
        </script>
    </body>
</html>
```

1.28 滑块 slider

作为一个拖拽式的交互性组件，滑块往往能给产品带来更好的操作体验。
模块加载名称：slider。

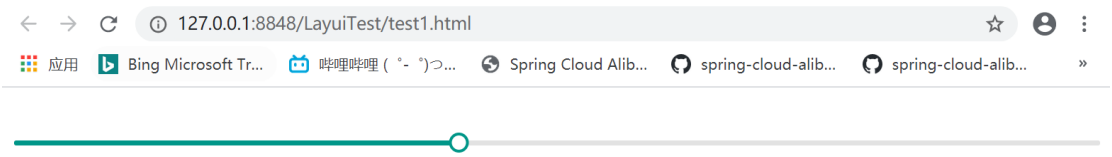
1.28.1 快速使用

通过对 slider 模块的使用，可以在页面构建出可拖动的滑动元素，如下代码是一个最基本的用法。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body style="padding: 10px;">
        <br />
        <br />
        <div id="slideTest1"></div>
        <script>
            var slider = layui.slider;
            //渲染
            slider.render({
                elem: '#slideTest1' //绑定元素
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.28.2 基础参数

支持的基础参数如图 1-xx 所示。

参数选项	说明	类型	默认值
elem	指向容器选择器	string/object	-
type	滑块类型，可选值有：*default*（水平滑块）、*vertical*（垂直滑块）	string	default
min	滑动条最小值，正整数，默认为 0	number	0
max	滑动条最大值	number	100
range	是否开启滑块的范围拖拽，若设为 true，则滑块将出现两个可拖拽的环	boolean	false
value	滑块初始值，默认为数字，若开启了滑块为范围拖拽（即 range: true），则需赋值数组，异表示开始和结尾的区间，如：value: [30, 60]	number/Array	0
step	拖动的步长	number	1
showstep	是否显示间断点	boolean	false
tips	是否显示文字提示	boolean	true
input	是否显示输入框（注意：若 range 参数为 true 则强制无效） 点击输入框的上下按钮，以及输入任意数字后回车或失去焦点，均可动态改变滑块	boolean	false
height	滑动条高度，需配合 type:"vertical" 参数使用	number	200
disabled	是否将滑块禁用拖拽	boolean	false
theme	主题颜色，以便用在不同的主题风格下	string	#009688

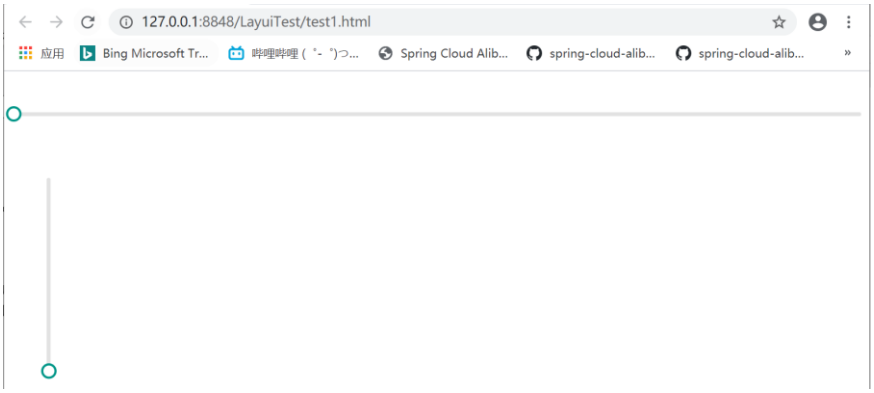
1.28.3 属性 type

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body style="padding: 10px;">
        <br />
        <br />
        <div id="slideTest1"></div>
        <br />
        <br />
        <br />
        <br />
        <div id="slideTest2"></div>
```

```
        <script>
            var slider = layui.slider;
            slider.render({
                elem: '#slideTest1',
                type: "default"
            });
            slider.render({
                elem: '#slideTest2',
                type: "vertical"
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.28.4 属性 min 和 max

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body style="padding: 10px;">
        <br />
        <br />
        <div id="slideTest1"></div>
        <script>
```

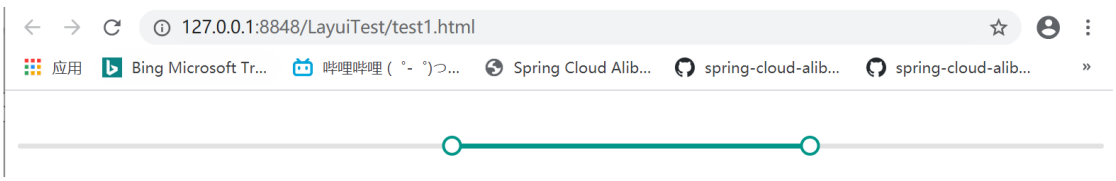
```
            var slider = layui.slider;
            slider.render({
                elem: '#slideTest1',
                min: 30,
                max: 90
            });
        </script>
    </body>
</html>
```

1.28.5 属性 range

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body style="padding: 10px;">
        <br />
        <br />
        <div id="slideTest1"></div>
        <script>
            var slider = layui.slider;
            slider.render({
                elem: '#slideTest1',
                range: true
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

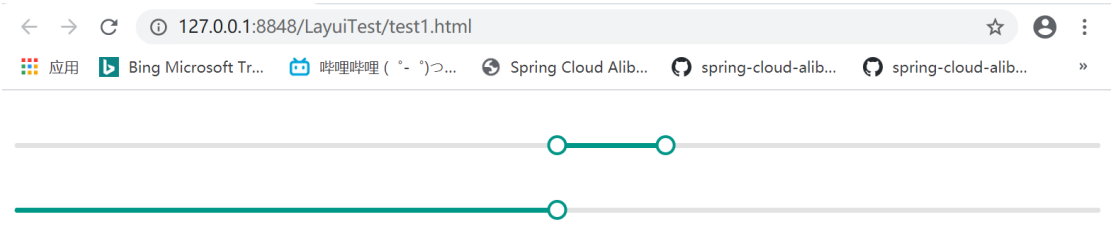


1.28.6 属性 value

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body style="padding: 10px;">
        <br />
        <br />
        <div id="slideTest1"></div>
        <br />
        <br />
        <br />
        <div id="slideTest2"></div>
        <script>
            var slider = layui.slider;
            slider.render({
                elem: '#slideTest1',
                range: true,
                value: [50, 60]
            });
            slider.render({
                elem: '#slideTest2',
                value: 50
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.28.7 属性 step

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body style="padding: 10px;">
        <br />
        <br />
        <div id="slideTest1"></div>
        <script>
            var slider = layui.slider;
            slider.render({
                elem: '#slideTest1',
                step: 10
            });
        </script>
    </body>
</html>
```

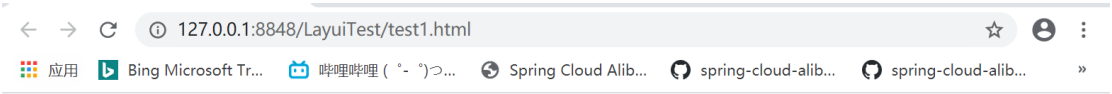
1.28.8 属性 showstep

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body style="padding: 10px;">
        <br />
        <br />
        <div id="slideTest1"></div>
        <script>
```

```
            var slider = layui.slider;
            slider.render({
                elem: '#slideTest1',
                step: 10,
                showstep: true
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.28.9 属性 tips

测试代码如下：

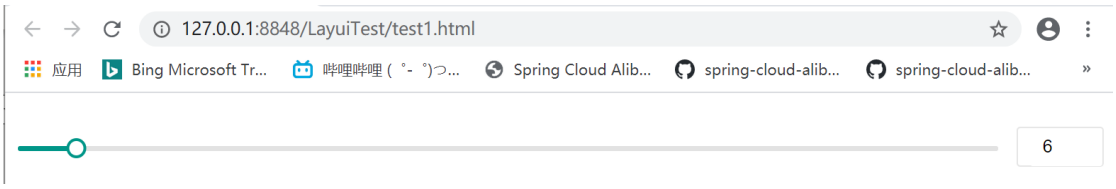
```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body style="padding: 10px;">
        <br />
        <br />
        <div id="slideTest1"></div>
        <script>
            var slider = layui.slider;
            slider.render({
                elem: '#slideTest1',
                tips: false
            });
        </script>
    </body>
</html>
```

1.28.10 属性 input

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body style="padding: 10px;">
        <br />
        <br />
        <div id="slideTest1"></div>
        <script>
            var slider = layui.slider;
            slider.render({
                elem: '#slideTest1',
                input: true
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.28.11 属性 height

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
```

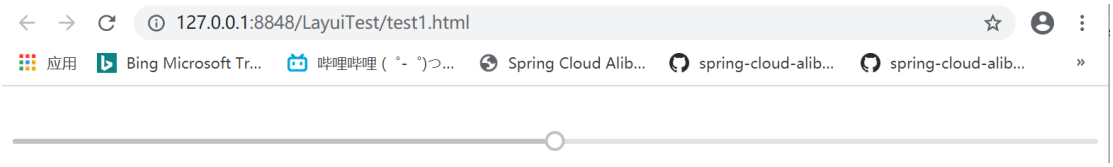
```
        <script src="layui/layui.all.js"></script>
    </head>
    <body style="padding: 10px;">
        <br />
        <br />
        <div id="slideTest1"></div>
        <script>
            var slider = layui.slider;
            slider.render({
                elem: '#slideTest1',
                type: "vertical",
                height: 500
            });
        </script>
    </body>
</html>
```

1.28.12 属性 disabled

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body style="padding: 10px;">
        <br />
        <br />
        <div id="slideTest1"></div>
        <script>
            var slider = layui.slider;
            slider.render({
                elem: '#slideTest1',
                value: 50,
                disabled: true
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

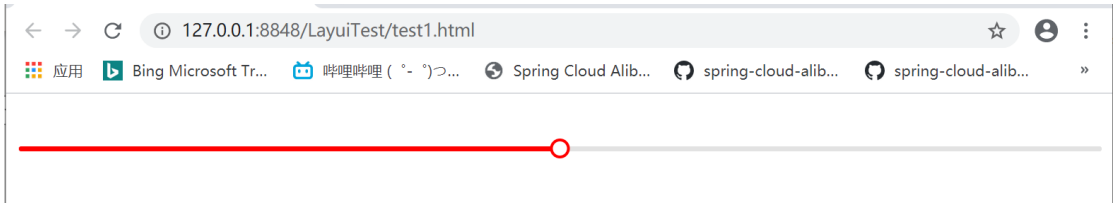


1.28.13 属性 theme

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body style="padding: 10px;">
        <br />
        <br />
        <div id="slideTest1"></div>
        <script>
            var slider = layui.slider;
            slider.render({
                elem: '#slideTest1',
                value: 50,
                theme: "red"
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



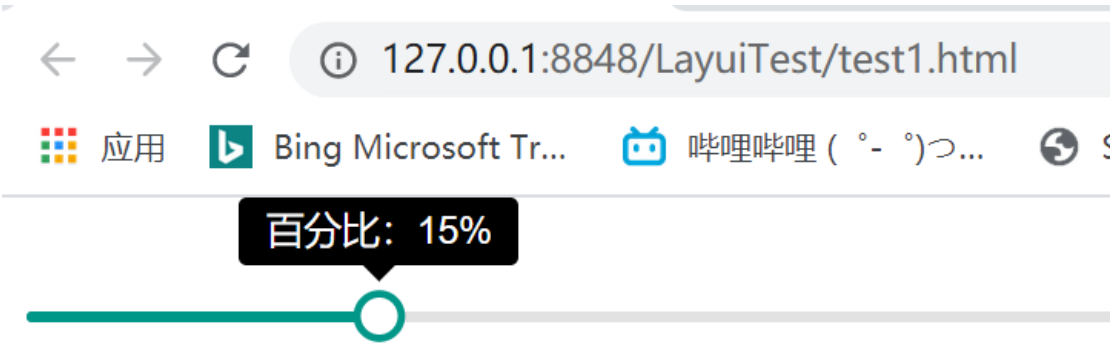
1.28.14 自定义提示文本

当鼠标放在圆点和滑块拖拽时均会触发提示层。其默认显示的文本是它的对应数值，也可以自定义提示内容。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body style="padding: 10px;">
        <br />
        <br />
        <div id="slideTest1"></div>
        <script>
            var slider = layui.slider;
            slider.render({
                elem: '#slideTest1',
                setTips: function(value) { //自定义提示文本
                    return "百分比：" + value + '%';
                }
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.28.15 数值改变的回调

在滑块数值被改变时触发。该回调非常重要，可动态获得滑块当前的数值。可以将得到的数值赋值给隐藏域或者进行一些其它操作。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body style="padding: 10px;">
        <br />
        <br />
        <div id="slideTest1"></div>
        <br />
        <br />
        <br />
        <div id="slideTest2"></div>
        <script>
            var slider = layui.slider;
            //当滑块为普通模式，回调返回的value是一个数值
            slider.render({
                elem: '#slideTest1',
                change: function(value) {
                    console.log(value) //动态获取滑块数值
                    //do something
                }
            });

            //当滑块为范围模式，回调返回的value是一个数组，包含开始和结尾
            slider.render({
                elem: '#slideTest2',
                range: true,
                change: function(value) {
                    console.log(value[0]) //得到开始值
                    console.log(value[1]) //得到结尾值
                    //do something
                }
            });
```

```
        </script>
    </body>
</html>
```

1.28.16 实例方法

执行 slider 实例时会返回一个当前实例的对象，里面包含针对当前实例的方法和属性。
语法：var ins1 = slider.render(options);
实例方法和属性如图 1-xx 所示。

实例方法和属性

```
var ins1 = slider.render(options); //获得实例对象

ins1.config //获得当前实例的配置项
ins1.setValue(nums); //动态给滑块赋值

```

1.28.17 动态改变滑块数值

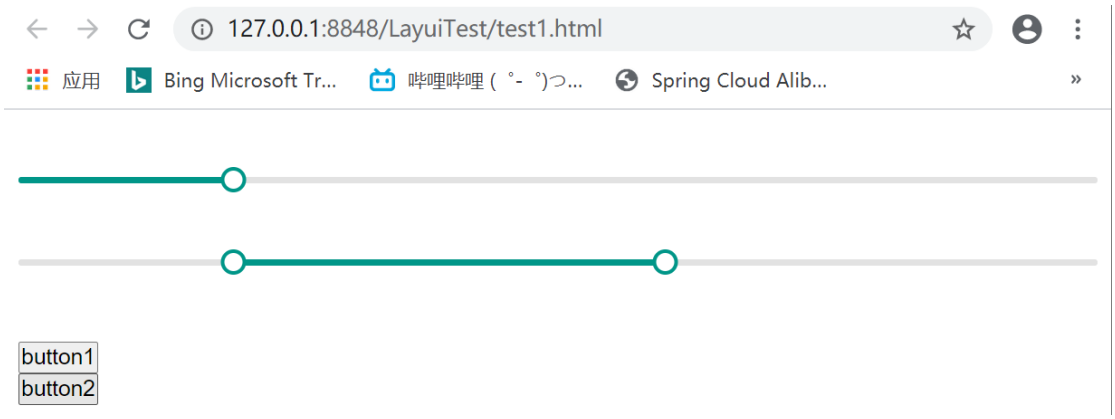
可以通过外部方法动态改变滑块数值。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body style="padding: 10px;">
        <br />
        <br />
        <div id="slideTest1"></div>
        <br />
        <br />
        <br />
        <div id="slideTest2"></div>
```

```
        <br />
        <br />
        <br />
        <input type="button" value="button1" id="button1">
        <br />
        <input type="button" value="button2" id="button2">
        <script>
            var slider = layui.slider;
            //当滑块为普通模式，回调返回的value是一个数值
            var ins1 = slider.render({
                elem: '#slideTest1'
            });

            //当滑块为范围模式，回调返回的value是一个数组，包含开始和结尾
            var ins2 = slider.render({
                elem: '#slideTest2',
                range: true
            });

            $(document).ready(function() {
                $("#button1").click(function() {
                    ins1.setValue(20)
                });
                $("#button2").click(function() {
                    //若滑块开启了范围（range: true）
                    ins2.setValue(20, 0) //设置开始值
                    ins2.setValue(60, 1) //设置结尾值
                });
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.29 评分组件 rate

rate 评分组件在电商和 O2O 平台尤为常见，一般用于对商家进行服务满意度评价。
模块加载名称：rate。

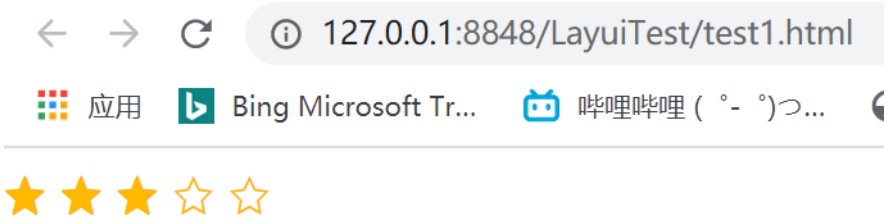
1.29.1 快速使用

rate 组件可以用来进行展示或评价，只需要通过更改参数设定来开启想要的功能。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var rate = layui.rate;
            //渲染
            var ins1 = rate.render({
                elem: '#test1' //绑定元素
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.29.2 基础参数

支持的基础参数如图 1-xx 所示。

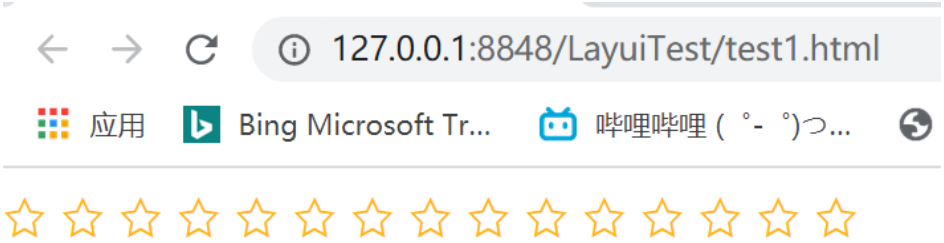
参数选项	说明	类型	默认值
elem	指向容器选择器	string/object	-
length	评分组件中具体星星的个数。个数当然是整数啦，残缺的星星很可怜的，所以设置了小数点的组件我们会默认向下取整	number	5
value	评分的初始值	number	0
theme	主题颜色。我们默认的组件颜色是 *#FFB800*，你可以根据自身喜好来更改组件的颜色，以适用不同场景	string	#FFB800
half	设定组件是否可以选择半星	boolean	false
text	是否显示评分对应的内容	boolean	false
readonly	是否只读，即只用于展示而不可点	boolean	false

1.29.3 属性 length

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var rate = layui.rate;
            //渲染
            var ins1 = rate.render({
                elem: '#test1',
                length: 15
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

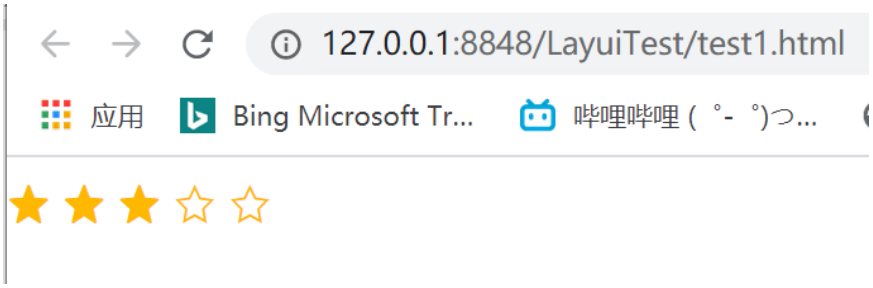


1.29.4 属性 value

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var rate = layui.rate;
            //渲染
            var ins1 = rate.render({
                elem: '#test1',
                value: 3
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

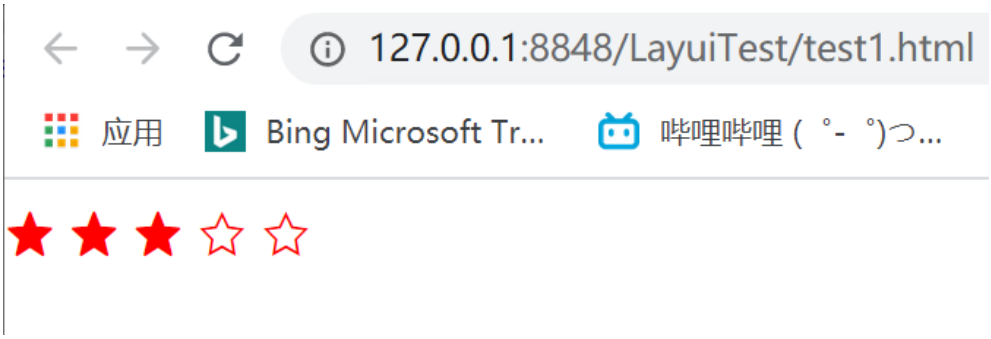


1.29.5 属性 theme

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var rate = layui.rate;
            //渲染
            var ins1 = rate.render({
                elem: '#test1',
                theme: 'red'
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



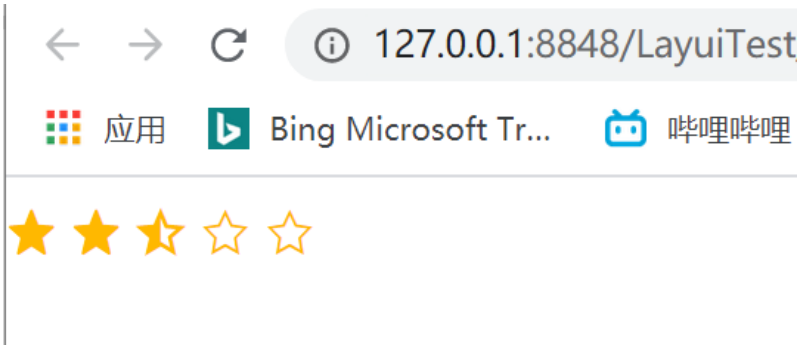
1.29.6 属性 half

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
```

```
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var rate = layui.rate;
            //渲染
            var ins1 = rate.render({
                elem: '#test1',
                half: true
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



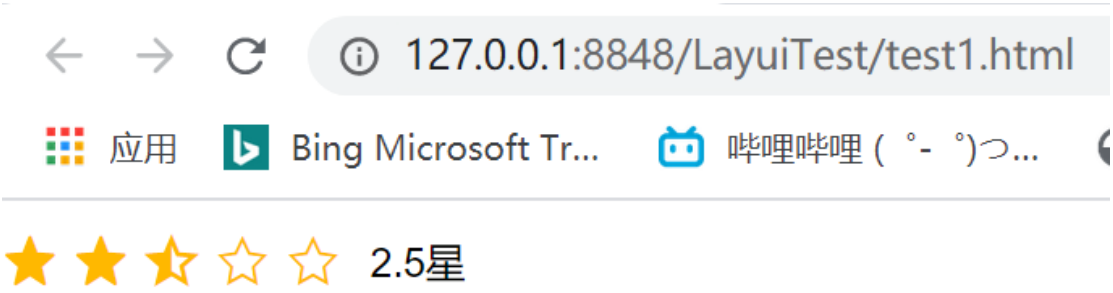
1.29.7 属性 text

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
```

```
            var rate = layui.rate;
            //渲染
            var ins1 = rate.render({
                elem: '#test1',
                half: true,
                text: true
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.29.8 属性 readonly

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var rate = layui.rate;
            //渲染
            var ins1 = rate.render({
                elem: '#test1',
                readonly: true,
                value: 3
```

```
            });
        </script>
    </body>
</html>
```

1.29.9 分数设置

分数设置规则如图 1-xx 所示。

如若你设置分数，我们会根据你是否开启半星功能，来做一个具体的规范：

关闭半星功能：

- *小数值大于 0.5 ： 分数向上取整，如 3.6 分，则系统自动更改为 4 分*
- *小数值小于等于 0.5 ： 分数向下取整，如 3.2 分，则系统自动更改为 3 分*
- *如果在关闭半星功能的情况下开启了文本，你会发现你的分数也相应的变成了整数*

开启半星功能：

- *不论你的小数值是 0.1 还是 0.9，都统一规划为 0.5，在文本开启的情况下，你可以看见你的分数并没有发生变化*

1.29.10 自定义文本的回调

通过 setText 函数，在组件初次渲染和点击后时产生回调。默认文本以星级显示，可以根据自己设定的文字来替换默认文本，如 “讨厌”或者“喜欢”。若用户选择分数而没有设定对应文字的情况下，系统会使用默认的文本。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var rate = layui.rate;
            //渲染
            var ins1 = rate.render({
                elem: '#test1',
                text: true,
                setText: function(value) {
```

```
                var arrs = {
                    '1': '极差',
                    '2': '差',
                    '3': '中等',
                    '4': '好',
                    '5': '非常好'
                };
                this.span.text(arrs[value] || (value + "星"));
            }
        });
    </script>
</body>
</html>
```

1.29.11 点击产生的回调

通过 choose 实现函数，在组件被点击后触发，回调分数，用户可根据分数来设置效果，比如出现弹出层。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test1"></div>
        <script>
            var rate = layui.rate;
            //渲染
            var ins1 = rate.render({
                elem: '#test1',
                choose: function(value) {
                    if (value > 4) alert('么么哒');
                }
            });
        </script>
    </body>
</html>
```

1.30 轮播组件 carousel

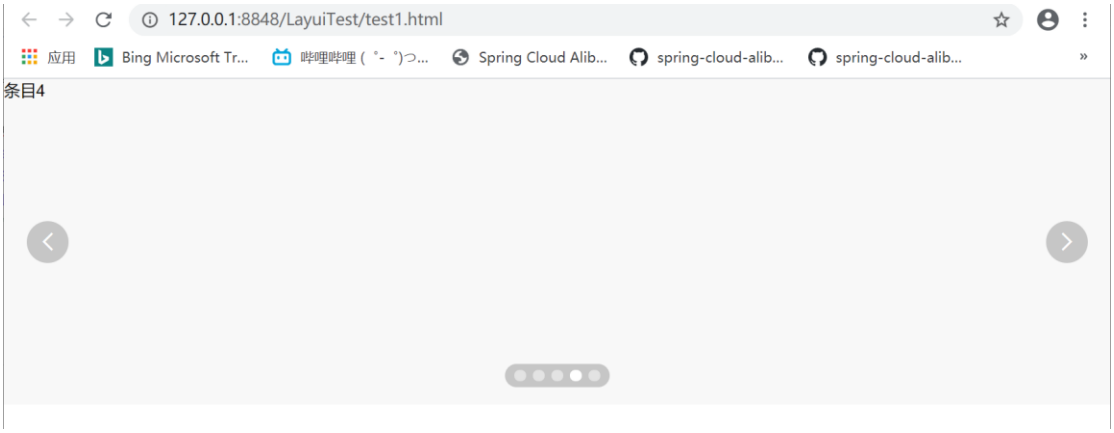
carousel 主要适用于跑马灯/轮播等交互场景。
模块加载名称：carousel。

1.30.1 快速使用

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-carousel" id="test1">
            <div carousel-item>
                <div>条目1</div>
                <div>条目2</div>
                <div>条目3</div>
                <div>条目4</div>
                <div>条目5</div>
            </div>
        </div>
        <!-- 条目中可以是任意内容，如：<img src=""> -->
        <script>
            var carousel = layui.carousel;
            //建造实例
            carousel.render({
                elem: '#test1',
                width: '100%', //设置容器宽度
                arrow: 'always' //始终显示箭头
                //,anim: 'updown' //切换动画方式
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



在 HTML 结构中，只需要简单地注意这两项：

（1）外层元素的 class="layui-carousel"用来标识为一个轮播容器

（2）内层元素的属性 carousel-item 用来标识条目

而 id 则用于 carousel 模块建造实例的元素指向，剩下的工作，就是按照实际需求给方法设置不同的基础参数了。

1.30.2 基础参数

通过核心方法：

carousel.render(options);

来对轮播设置基础参数，也可以通过方法：

carousel.set(options);

来设定全局基础参数。

支持的基础参数如图 1-xx 所示。

可选项	说明	类型	默认值
elem	指向容器选择器，如：elem: '#id'。也可以是DOM对象	string/object	无
width	设定轮播容器宽度，支持像素和百分比	string	'600px'
height	设定轮播容器高度，支持像素和百分比	string	'280px'
full	是否全屏轮播	boolean	false
anim	轮播切换动画方式，可选值为： • default（左右切换） • updown（上下切换） • fade（渐隐渐显切换）	string	'default'
autoplay	是否自动切换	boolean	true
interval	自动切换的时间间隔，单位：ms（毫秒），不能低于800	number	3000
index	初始开始的条目索引	number	0
arrow	切换箭头默认显示状态，可选值为： • hover（悬停显示） • always（始终显示） • none（始终不显示）	string	'hover'
indicator	指示器位置，可选值为： • inside（容器内部） • outside（容器外部） • none（不显示） 注意：如果设定了 *anim:'updown'*，该参数将无效	string	'inside'
trigger	指示器的触发事件	string	'click'

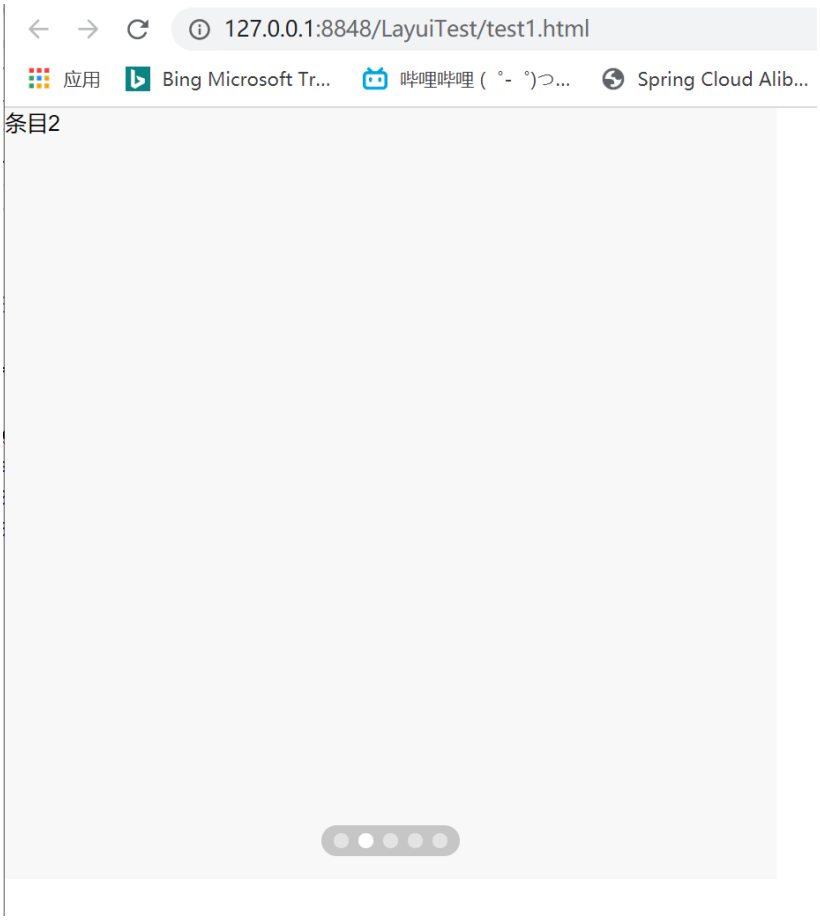
1.30.3 属性 width 和 height

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-carousel" id="test1">
            <div carousel-item>
                <div>条目1</div>
                <div>条目2</div>
                <div>条目3</div>
                <div>条目4</div>
```

```
            <div>条目5</div>
        </div>
    </div>
    <!-- 条目中可以是任意内容，如：<img src=""> -->
    <script>
        var carousel = layui.carousel;
        //建造实例
        carousel.render({
            elem: '#test1',
            width: '500px',
            height: "500px"
        });
    </script>
</body>
</html>
```

运行效果如图 1-xx 所示。



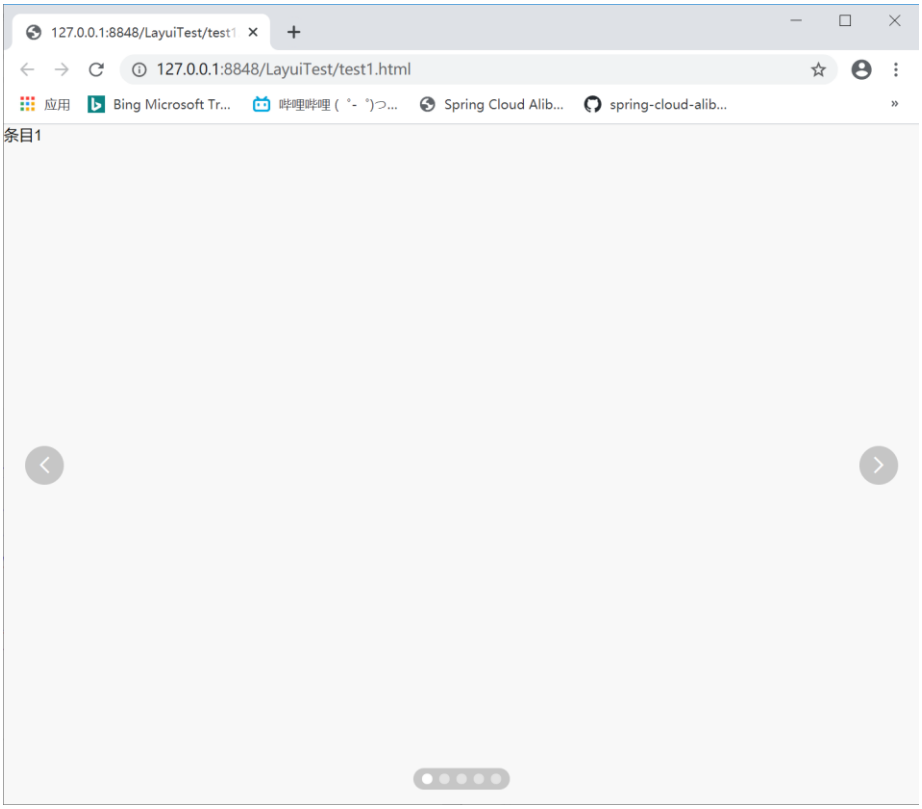
1.30.4 属性 full

测试代码如下：

```
<!DOCTYPE html>
```

```
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-carousel" id="test1">
            <div carousel-item>
                <div>条目1</div>
                <div>条目2</div>
                <div>条目3</div>
                <div>条目4</div>
                <div>条目5</div>
            </div>
        </div>
        <!-- 条目中可以是任意内容，如：<img src=""> -->
        <script>
            var carousel = layui.carousel;
            //建造实例
            carousel.render({
                elem: '#test1',
                full: true
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.30.5 属性 anim

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-carousel" id="test1">
            <div carousel-item>
                <div>条目1</div>
                <div>条目2</div>
                <div>条目3</div>
                <div>条目4</div>
                <div>条目5</div>
            </div>
        </div>
```

```
        <!-- 条目中可以是任意内容，如：<img src=""> -->
        <script>
            var carousel = layui.carousel;
            //建造实例
            carousel.render({
                elem: '#test1',
                //anim: "updown" //上下切换
                //anim: "fade" //渐隐渐显切换
                anim: "default" //default（左右切换）
            });
        </script>
    </body>
</html>
```

1.30.6 属性 autoplay

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-carousel" id="test1">
            <div carousel-item>
                <div>条目1</div>
                <div>条目2</div>
                <div>条目3</div>
                <div>条目4</div>
                <div>条目5</div>
            </div>
        </div>
        <!-- 条目中可以是任意内容，如：<img src=""> -->
        <script>
            var carousel = layui.carousel;
            //建造实例
            carousel.render({
                elem: '#test1',
                autoplay: false
```

```
            });
        </script>
    </body>
</html>
```

1.30.7 属性 interval

测试代码如下：

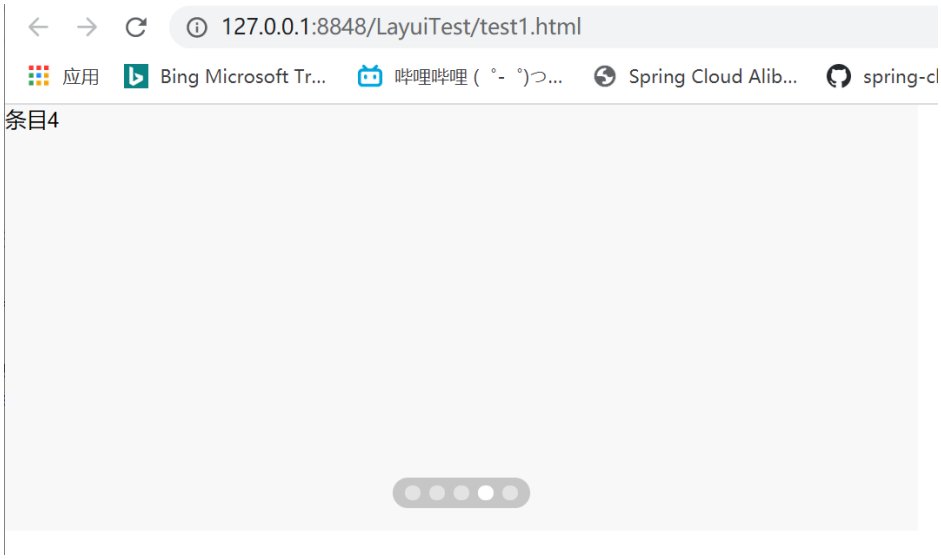
```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-carousel" id="test1">
            <div carousel-item>
                <div>条目1</div>
                <div>条目2</div>
                <div>条目3</div>
                <div>条目4</div>
                <div>条目5</div>
            </div>
        </div>
        <!-- 条目中可以是任意内容，如：<img src=""> -->
        <script>
            var carousel = layui.carousel;
            //建造实例
            carousel.render({
                elem: '#test1',
                interval: 1000
            });
        </script>
    </body>
</html>
```

1.30.8 属性 index

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-carousel" id="test1">
            <div carousel-item>
                <div>条目1</div>
                <div>条目2</div>
                <div>条目3</div>
                <div>条目4</div>
                <div>条目5</div>
            </div>
        </div>
        <!-- 条目中可以是任意内容，如：<img src=""> -->
        <script>
            var carousel = layui.carousel;
            //建造实例
            carousel.render({
                elem: '#test1',
                index: 3
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.30.9 属性 arrow

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-carousel" id="test1">
            <div carousel-item>
                <div>条目1</div>
                <div>条目2</div>
                <div>条目3</div>
                <div>条目4</div>
                <div>条目5</div>
            </div>
        </div>
        <!-- 条目中可以是任意内容，如：<img src=""> -->
        <script>
            var carousel = layui.carousel;
            //建造实例
            carousel.render({
                elem: '#test1',
```

```
            //arrow: "hover" //悬停显示
            //arrow: "always" //始终显示
            arrow: "none" //始终不显示
        });
    </script>
</body>
</html>
```

1.30.10 属性 indicator

注意：如果设定了 anim:'updown'，该参数将无效。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-carousel" id="test1">
            <div carousel-item>
                <div>条目1</div>
                <div>条目2</div>
                <div>条目3</div>
                <div>条目4</div>
                <div>条目5</div>
            </div>
        </div>
        <!-- 条目中可以是任意内容，如：<img src=""> -->
        <script>
            var carousel = layui.carousel;
            //建造实例
            carousel.render({
                elem: '#test1',
                //indicator: "inside" //容器内部
                //indicator: "outside" //容器外部
                indicator: "none" //不显示
            });
        </script>
```

```
    </body>
</html>
```

1.30.11 属性 trigger

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-carousel" id="test1">
            <div carousel-item>
                <div>条目1</div>
                <div>条目2</div>
                <div>条目3</div>
                <div>条目4</div>
                <div>条目5</div>
            </div>
        </div>
        <!-- 条目中可以是任意内容，如：<img src=""> -->
        <script>
            var carousel = layui.carousel;
            //建造实例
            carousel.render({
                elem: '#test1',
                trigger: "mouseover"
            });
        </script>
    </body>
</html>
```

1.30.12 切换事件

轮播的每一次切换时触发，回调函数返回一个 object 参数。

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-carousel" id="test1" lay-filter="mycarousel">
            <div carousel-item>
                <div>条目1</div>
                <div>条目2</div>
                <div>条目3</div>
                <div>条目4</div>
                <div>条目5</div>
            </div>
        </div>
        <!-- 条目中可以是任意内容，如：<img src=""> -->
        <script>
            var carousel = layui.carousel;
            //建造实例
            carousel.render({
                elem: '#test1',
            });

            //监听轮播切换事件
            carousel.on('change(mycarousel)', function(obj) { //test1来源
于对应HTML容器的 lay-filter="test1" 属性值
                console.log(obj.index); //当前条目的索引
                console.log(obj.prevIndex); //上一个条目的索引
                console.log(obj.item); //当前条目的元素对象
            });
        </script>
    </body>
</html>
```

1.30.13 重载方法

事实上，在执行 carousel.render(options)方法时，有返回一个当前实例的对象。该对象包含了用于操作当前轮播的一些属性和方法，示例代码如图 1-xx 所示。

```
code
01. var ins = carousel.render(options);
02.
03. //重置轮播
04. ins.reload(options);
05.
```

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div class="layui-carousel" id="test1" lay-filter="mycarousel">
            <div carousel-item>
                <div>条目1</div>
                <div>条目2</div>
                <div>条目3</div>
                <div>条目4</div>
                <div>条目5</div>
            </div>
        </div>
        <!-- 条目中可以是任意内容，如：<img src=""> -->
        <script>
            var carousel = layui.carousel;
            //建造实例
            var mycarousel = carousel.render({
                elem: '#test1',
            });

            $(document).ready(function() {
                setTimeout(function() {
                    mycarousel.reload({
                        interval: 1000,
```

```
                });
            }, 5000);
        });
    </script>
</body>
</html>
```

1.31 工具集 util

将一些工具性元素放入 util 模块中，以供选择性使用。其内部由多个小工具组件组成，他们也许不是必须的，但很多时候却能为页面提供良好的辅助作用。

模块加载名称：util。

1.31.1 固定块

固定块通常会出现在固定位置，由两个可选的 bar 和一个默认必选的 TopBar 组成。

语法：util.fixbar(options)，其中参数 options 是一个对象，可支持的 key 如图 1-xx 所示。

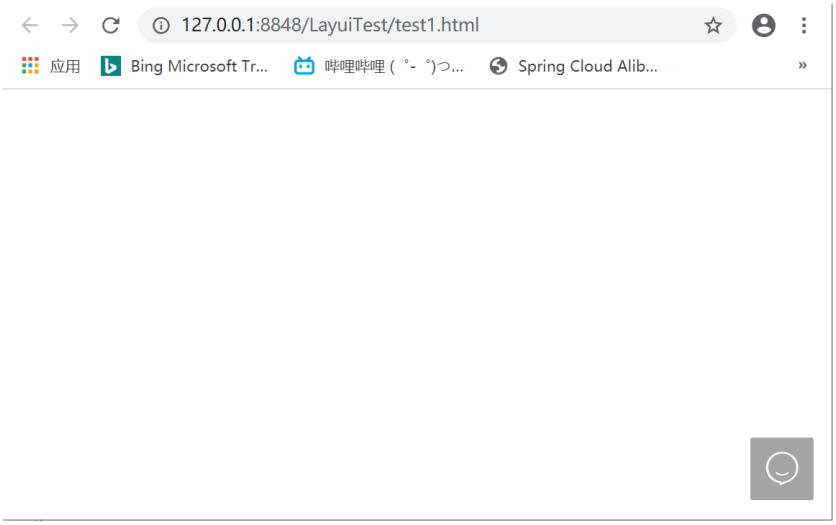
参数	类型	描述
bar1	Boolean/String	默认false。如果值为true，则显示第一个bar，带有一个默认图标。如果值为图标字符，则显示第一个bar，并覆盖默认图标
bar2	Boolean/String	默认false。如果值为true，则显示第二个bar，带有一个默认图标。如果值为图标字符，则显示第二个bar，并覆盖默认图标
bgcolor	String	自定义区块背景色
showHeight	Number	用于控制出现TOP按钮的滚动条高度临界值。默认：200
css	Object	你可以通过重置bar的位置，比如 css: {right: 100, bottom: 100}
click	Function	点击bar的回调，函数返回一个type参数，用于区分bar类型。支持的类型有：bar1、bar2、top

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <script>
```

```
            var util = layui.util;
            //执行
            util.fixbar({
                bar1: true,
                click: function(type) {
                    console.log(type);
                    if (type === 'bar1') {
                        alert('点击了bar1')
                    }
                }
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



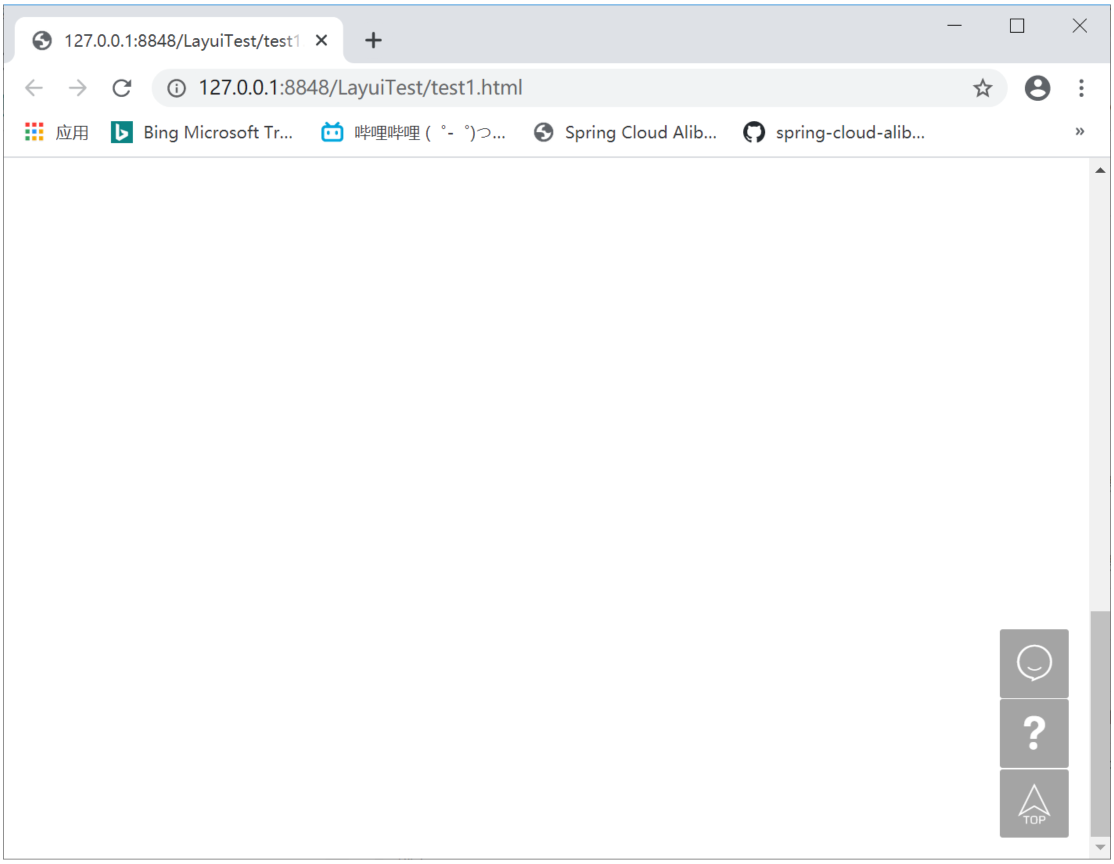
1.31.1.1 属性 bar1 和 bar2 值为 true

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
```

```
        <br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br />
        <script>
            var util = layui.util;
            //执行
            util.fixbar({
                bar1: true,
                bar2: true,
                click: function(type) {
                    console.log(type);
                    if (type === 'bar1') {
                        alert('点击了bar1')
                    }
                    if (type === 'bar2') {
                        alert('点击了bar2')
                    }
                }
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



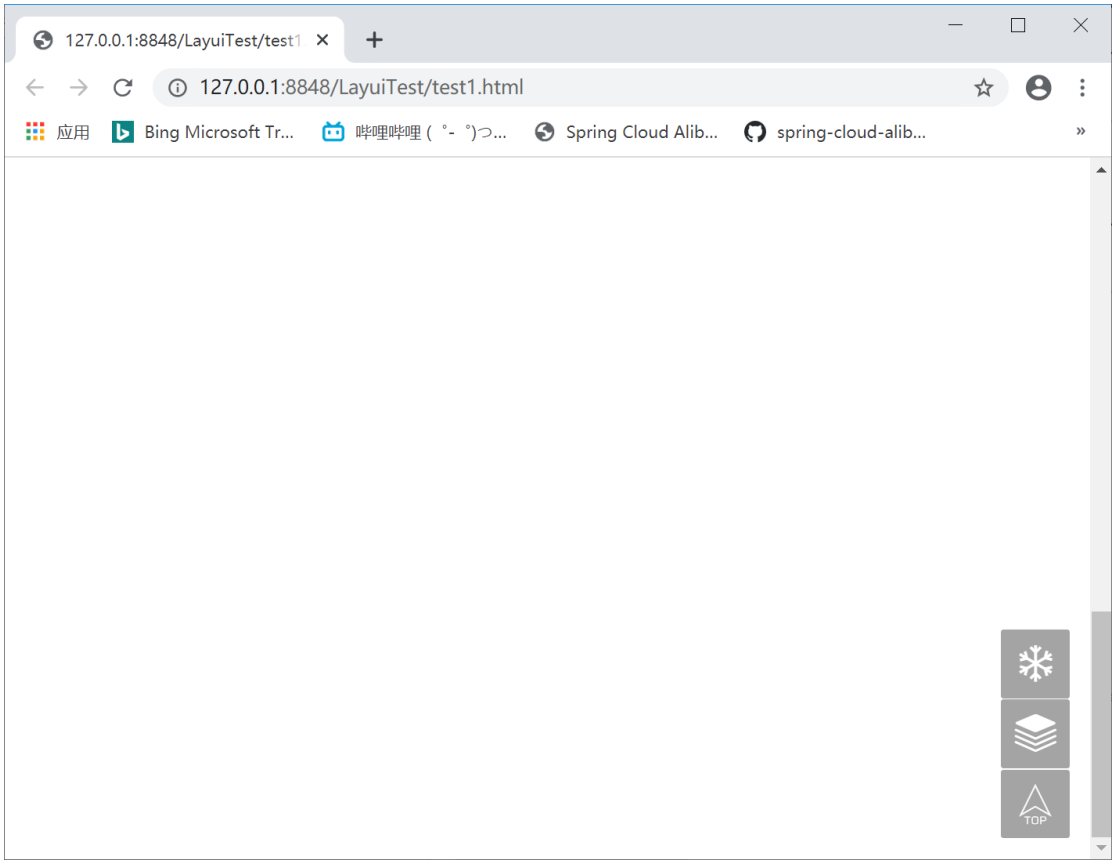
1.31.1.2 属性 bar1 和 bar2 值为图标字符

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br />
```

```
        <script>
            var util = layui.util;
            //执行
            util.fixbar({
                bar1: "&#xe6b1;",
                bar2: "&#xe656;",
                click: function(type) {
                    console.log(type);
                    if (type === 'bar1') {
                        alert('点击了bar1')
                    }
                    if (type === 'bar2') {
                        alert('点击了bar2')
                    }
                }
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。

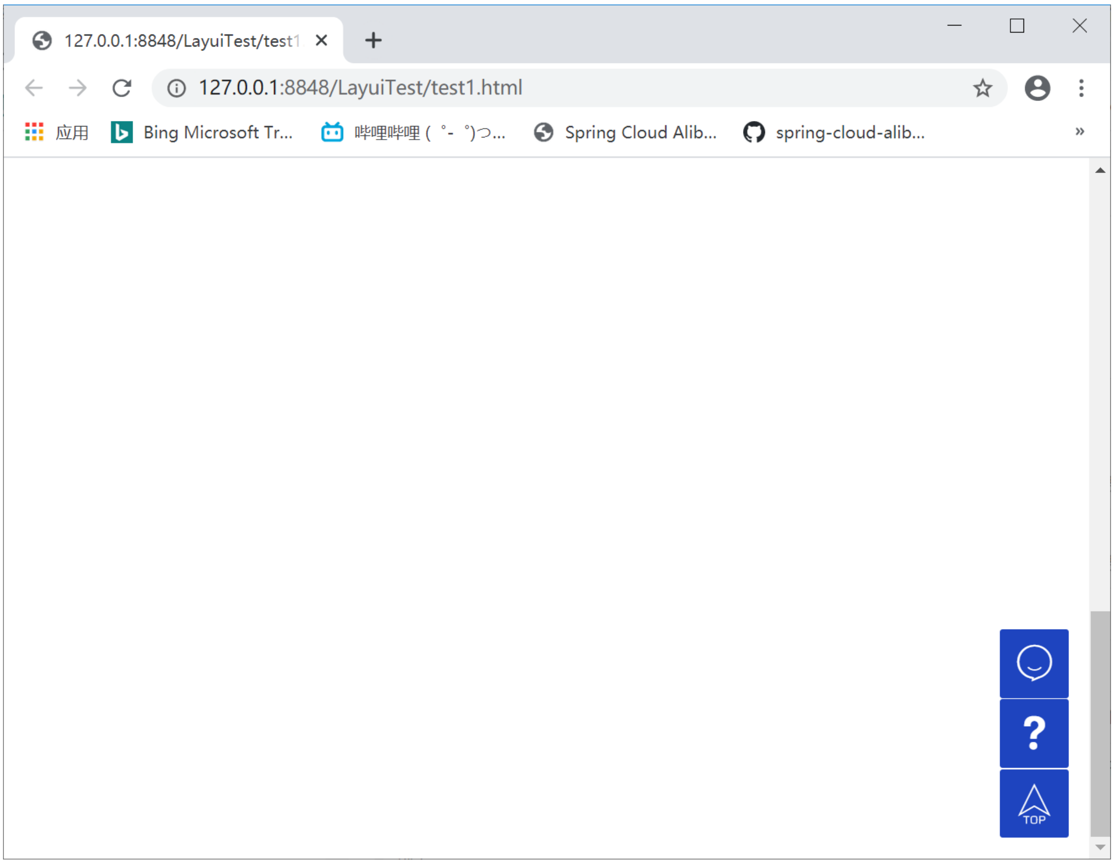


1.31.1.3 属性 bgcolor

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br />
        <script>
            var util = layui.util;
            //执行
            util.fixbar({
                bar1: true,
                bar2: true,
                bgcolor: "#123ABC",
                click: function(type) {
                    console.log(type);
                    if (type === 'bar1') {
                        alert('点击了bar1')
                    }
                    if (type === 'bar2') {
                        alert('点击了bar2')
                    }
                }
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.31.1.4 属性 showHeight

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br />
```

```
        <script>
            var util = layui.util;
            //执行
            util.fixbar({
                bar1: true,
                bar2: true,
                showHeight: 100,
                click: function(type) {
                    console.log(type);
                    if (type === 'bar1') {
                        alert('点击了bar1')
                    }
                    if (type === 'bar2') {
                        alert('点击了bar2')
                    }
                }
            });
        </script>
    </body>
</html>
```

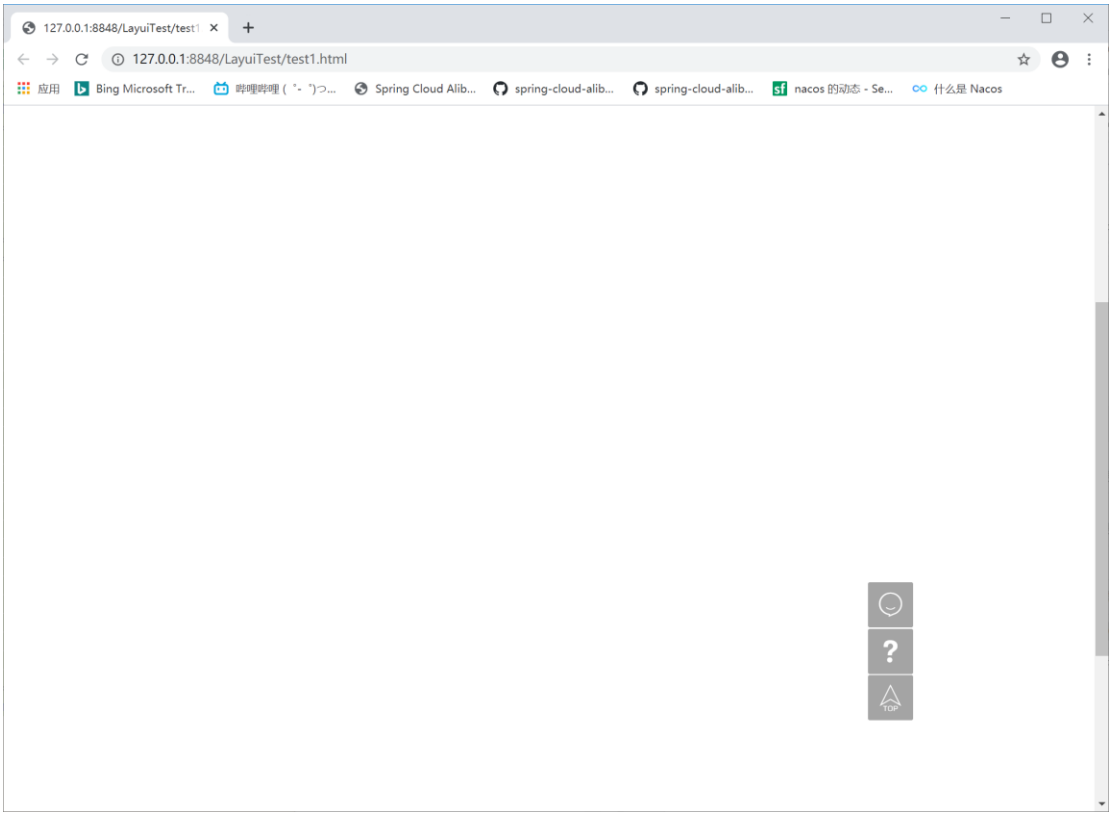
1.31.1.5 属性 css

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br /><br />
        <script>
```

```
            var util = layui.util;
            //执行
            util.fixbar({
                bar1: true,
                bar2: true,
                css: {
                    right: 200,
                    bottom: 100
                },
                click: function(type) {
                    console.log(type);
                    if (type === 'bar1') {
                        alert('点击了bar1')
                    }
                    if (type === 'bar2') {
                        alert('点击了bar2')
                    }
                }
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



1.31.2 倒计时

倒计时组件并不负责 UI 元素的呈现，而仅仅只是返回倒计时的数据，这意味着可以将它应用在任何倒计时相关的业务中。

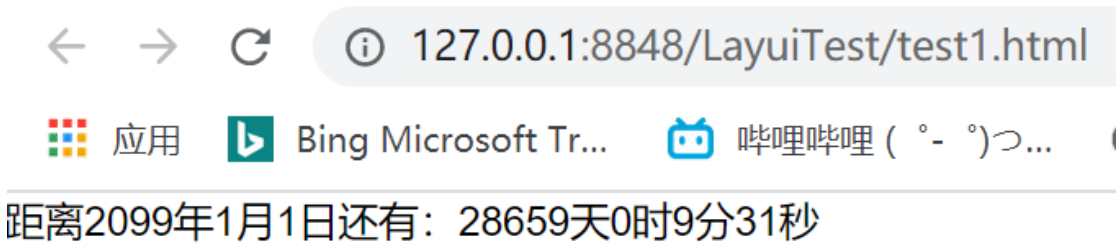
语法：util.countdown(endTime, serverTime, callback)，参数作用如图 1-xx 所示。

参数	说明
endTime	结束时间戳或Date对象，如：4073558400000，或：new Date(2099,1,1).
serverTime	当前服务器时间戳或Date对象
callback	回调函数。如果倒计时尚在运行，则每一秒都会执行一次。并且返回三个参数： *date*（包含天/时/分/秒的对象）、*serverTime*（当前服务器时间戳或Date对象），*timer*（计时器返回的ID值，用于clearTimeout）

测试代码如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <script src="js/jquery3.5.1.js"></script>
        <link rel="stylesheet" href="layui/css/layui.css" />
        <script src="layui/layui.all.js"></script>
    </head>
    <body>
        <div id="test"></div>
        <script>
            var util = layui.util;
            //示例
            var endTime = new Date(2099, 1, 1).getTime(); //假设为结束日期
            var serverTime = new Date().getTime(); //假设为当前服务器时间，
这里采用的是本地时间，实际使用一般是取服务端的
            util.countdown(endTime, serverTime, function(date, serverTime,
timer) {
                var str = date[0] + '天' + date[1] + '时' + date[2] + '分'
+ date[3] + '秒';
                layui.$('#test').html('距离2099年1月1日还有：' + str);
            });
        </script>
    </body>
</html>
```

运行效果如图 1-xx 所示。



其它参考资料网址如下：

https://www.cnblogs.com/xrxiaolong/articles/8073920.html

https://blog.csdn.net/qq_43349566/article/details/89467005

https://blog.csdn.net/molihuakai_118/article/details/92426747
https://www.bbsmax.com/A/Gkz1Q2pQzR/
http://www.idepy.com/749.html
https://blog.csdn.net/CaptainJava/article/details/102853088
https://blog.51cto.com/1197822/2472692
https://www.cnblogs.com/kstrive/p/8531945.html
https://www.it610.com/article/1288780306295365632.htm
https://www.zhangshengrong.com/p/3mNmm06zNj/
https://blog.csdn.net/weixin_37365539/article/details/106549789
https://blog.csdn.net/xkl923620865/article/details/88963033?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-2.channel_param&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-2.channel_param
https://zhuanlan.zhihu.com/p/64827997
https://blog.csdn.net/yang134679/article/details/98896090
https://blog.csdn.net/zhangxiaoyang0/article/details/78045403
https://www.bubaijun.com/page.php?id=182
https://www.cnblogs.com/myLeisureTime/p/12175491.html
https://www.cnblogs.com/xiaostudy/p/12335716.html
https://www.zhangshengrong.com/p/yOXD5zYj1B/
http://www.45fan.com/article.php?aid=19090634100294432115466297
https://www.cnblogs.com/xiaonangua/p/11491692.html
http://www.3672js.com/javascriptjc/31884.html
https://www.cnblogs.com/Renyi-Fan/p/10777642.html
https://blog.csdn.net/weixin_33805992/article/details/93572042
https://blog.csdn.net/weixin_39186924/article/details/89395374
https://blog.csdn.net/qq6759/article/details/102708569
https://www.cnblogs.com/zhinian-/p/12916601.html
https://www.jianshu.com/p/f200d8d505e8