

Vue 瀑布流踩坑记 (vue-waterfall-easy)

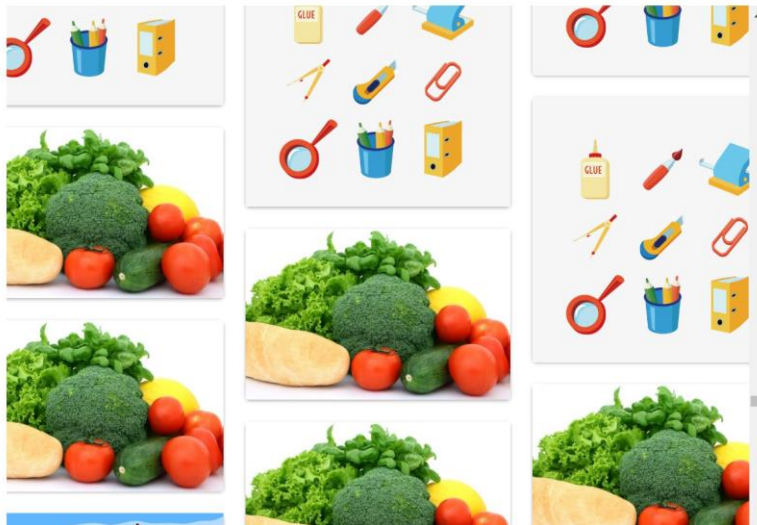
皓月

项目中间要展示一系列图片，做成瀑布流展示比较美观，且不用过多考虑图片的大小排列这个问题。通过网上查询得知了 vue-waterfall-easy 这个很棒的插件，部署简单，自动排列。

但尽管部署“简单”，但中间也碰到了一系列的小坑，在这里记录以帮助后面的人避坑。

插件 github 地址 <https://github.com/lfyfly/vue-waterfall-easy>

以下是实现的效果



安装

```
npm install vue-waterfall-easy --save
```

遇到的坑点:

1. 确保后端的数据正确情况下，图片在前端显示不出来

点开 inspect 发现整个块的高度是 0px，应当是在还没有拉取到数据时这个模块就已经渲染了，既然没有数据那么就是 0px 了，而又因为是 0px，所以拉取数据后也无法展示，形成了一个死锁。

因此将它的高度强制改为 500px

```
.vue-waterfall-easy-scroll {  
  height: 500px !important;  
}
```

2. 有两个滚动条

其中一个插件自带的，另一个是浏览器提供的。

而想想浏览器出现滚动条的原因在于当前页面超过了显示器的范围，所以只要让当前所排布的模块高度和小于能显示的范围，就可以把浏览器自带滚动条去除了。

```
.vue-waterfall-easy {  
  height: 600px !important;  
}
```

部署技巧

在完成一个功能时，一步一步来，大概可以分为这三步

1. 前端硬编码瀑布流展示
2. 从后端硬编码获取数据
3. 后端从数据库获取数据

这样的话方便定位错误

后记

发现基础扎实才是硬道理，大家碰到的问题都不一样，所以直接采用其它人的方法而不理解的话会导致后续一系列问题。

比如在最初碰到图片无法显示的时候，有个博主给出的是将 position 改为 fixed 的方法，这个方法确实可以让图片显示，原因在于这时候它的展示不依赖于父组件了。

但是这样改的直接后果导致了没有滚动条，因为滚动条是在界面展示不下的时候产生的，但是 fixed 这个属性它的展示是固定的。这个地方卡了我很久，在 github 的 issue 上有人说这是因为 @scrollReachBottom 这个事件大写，但是 html 大小写不敏感，因此 v-on 事件监听器在 DOM 模板中会被自动转换为全小写，无法激活事件。以为是这个问题，转而看插件 github 样例，发现引入了 pug，也跟着尝试 pug，发现依然有问题，总而言之，绕来绕去的。

最后是突然意识到这个 fixed 很奇怪，思考一下发觉问题在哪里了。

两个滚动条也是按照这个思路很快解决了。

所以，每一步代码都要理解了再往下完成。前面逻辑不清楚，只会导致后面一团糟。

总的代码（只有五十来行）

```
<template>
  <div class="container">
    <div id="content">
      <vue-waterfall-easy :imgsArr="imgsArr" @scrollReachBottom="getData"></vue-waterfall-easy>
    </div>
  </div>
</template>
```

```
<script>
  import vueWaterfallEasy from 'vue-waterfall-easy'
  export default {
    name: 'Words',
    data() {
      return {
        imgsArr: [],
        group: 0, //用于标记第几组图片的到来
      }
    },
    components: {
      vueWaterfallEasy
    },
    methods: {
      getData() { //从后端拉取数据
        const path = '/words'
        this.$axios.get(path)
          .then(response => {
            console.log(response)
            var arr = []
            for (var i = 0; i < response.data.length; i++) {
              arr.push({src: `static/img/${response.data[i]}.jpg`})
            }
            this.imgsArr = this.imgsArr.concat(arr)
            this.group++
            console.log(this.imgsArr)
          })
      }
    },
    created() {
      this.getData()
    }
  }
</script>
```

```
<style>

  .vue-waterfall-easy-scroll {
    height: 500px !important;
  }

  .vue-waterfall-easy {
    height: 600px !important;
  }

</style>
```