# 项目情况

颜色：1afa29

# 知识拓展

## Vue定时器与JS 定时器 setInterval() 和 setTimeout()

[https://mp.weixin.qq.com/s?src=11&timestamp=1613526621&ver=2895&signature=gflVcR9\*v-m3MGTXE02Vmas1W5tmtedarwT1MLSu00muoEamt2JRG-DgqK86UMkVCKIspO2gwksKLN-9HV68zGFTnp-FcLNq8sHa2bQSEQPiV100dnWjI9mbOJGDhfya&new=1](https://mp.weixin.qq.com/s?src=11&timestamp=1613526621&ver=2895&signature=gflVcR9*v-m3MGTXE02Vmas1W5tmtedarwT1MLSu00muoEamt2JRG-DgqK86UMkVCKIspO2gwksKLN-9HV68zGFTnp-FcLNq8sHa2bQSEQPiV100dnWjI9mbOJGDhfya&new=1)

在vue中，有两套定时器，一套是浏览器API，window对象上的；另一套就是vue/nodejs封装的，需要引入

import { setInterval, clearInterval } from 'timers'

建议使用window对象自带的，因为不容易错。 如果一不小心只引入一个，就怎么也清不掉了。

import { setInterval } from 'timers' *// 错误*

**一定要在beforeDestroy中清除定时器。**

data () { return { timer: 0 }},*//模块初始化时打开定时器*created () { this.timer = setInterval(() => { *//do something* *//定时器的回调函数中需要注意 this 指向* }, 5000)},*//销毁前清除定时器*beforeDestroy () { clearInterval(this.timer)}

js 定时器有以下两个方法：

setInterval() ：按照指定的周期（以毫秒计）来调用函数或计算表达式。方法会不停地调用函数，直到 clearInterval()  被调用或窗口被关闭。

setTimeout() ：在指定的毫秒数后调用函数或计算表达式。

**setInterval()**

语法

setInterval(code,millisec,lang)

参数 描述

code 必需。要调用的函数或要执行的代码串。

millisec 必须。周期性执行或调用 code 之间的时间间隔，以毫秒计。

lang 可选。 JScript | VBScript | JavaScript

以下实例在每 1000 毫秒执行 clock() 函数。实例中也包含了停止执行的按钮:

在需要重复发送请求或者某些效果的时候，一般都会想到使用setInterval，但是它的一些弊端，会给程序带来很大的隐患

**一、弊端**

setInterval对自己调用的代码是否报错漠不关心。即使调用的代码报错了，它依然会持续的调用下去。

setInterval无视网络延迟。在使用ajax轮询服务器是否有新数据时，必定会有一些人会使用setInterval，然而无论网络状况如何，它都会去一遍又一遍的发送请求，如果网络状况不良，一个请求发出，还没有返回结果，它会坚持不懈的继续发送请求，最后导致的结果就是请求堆积。

setInterval并不定时。如果它调用的代码执行的时间小于定时的时间，它会跳过调用，这就导致无法按照你需要的执行次数或无法得到你想要的结果。

**二、解决方案**使用setTimeout代替setInterval。 可以给setTimeout设置时间后，在最后调用自身。如果希望“匀速”触发。可以计算代码执行时间，用希望的延迟减去上次执行的时间。

注：有一种想法是将setInterval的延迟时间设置的长于上述的几种时间，来达到绝对的均速调用。但事实上，js的计时器因为自身机制的原因，存在4ms–15ms的误差。

**setTimeout()**语法

setTimeout(code,millisec,lang)

参数 描述

code 必需。要调用的函数后要执行的 JavaScript 代码串。

millisec 必需。在执行代码前需等待的毫秒数。

lang 可选。脚本语言可以是：JScript | VBScript | JavaScript

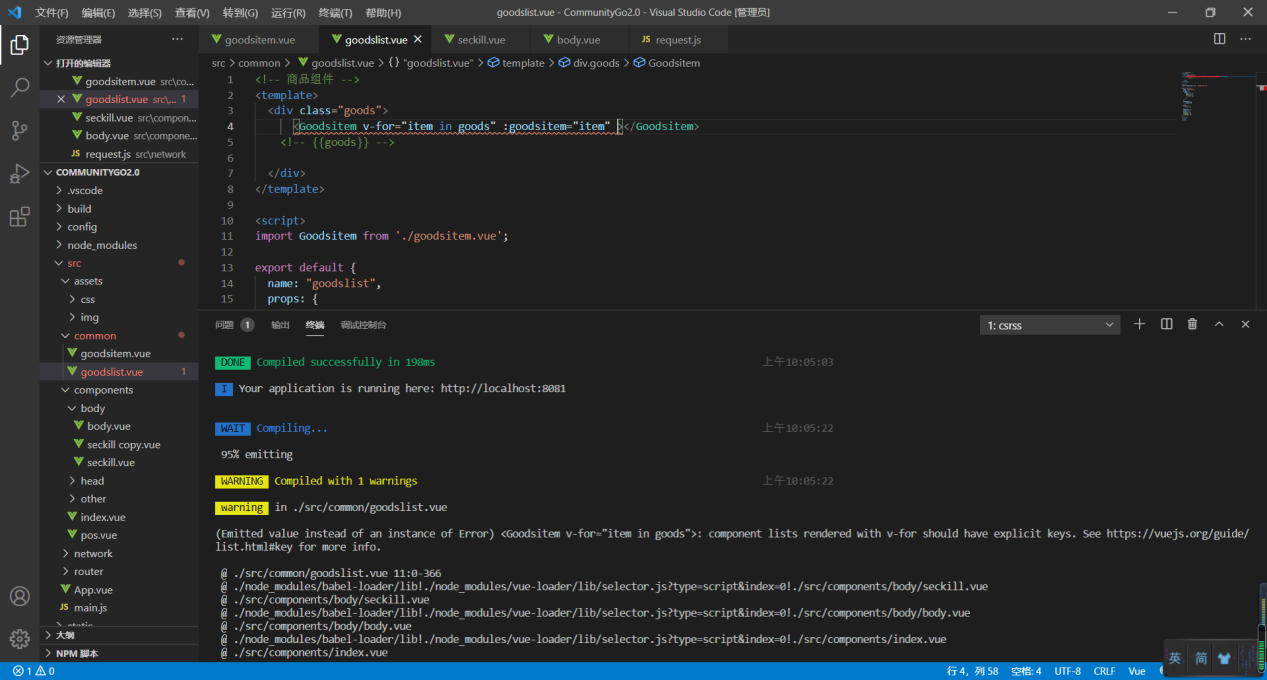
如果想要在一个函数中启用定时器  又想在另一个函数关闭这个函数 可以这样做:

var timer1 = null;  
function start(){ if ( timer1 ) return; timer1 = setInterval("test()",200);}function end(){ if ( timer1 ) { clearInterval(timer1); timer1 = null; }}

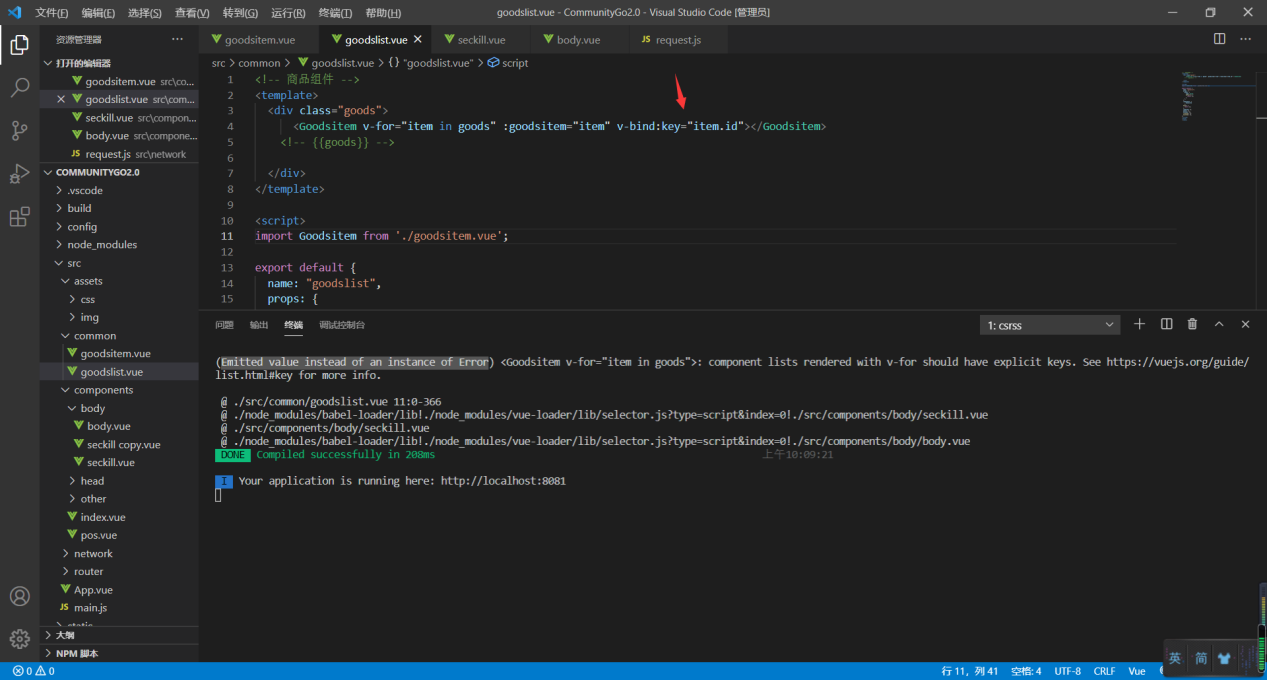
注意事项: 这里的 timer1 相当于setInterval 的 id, 执行clearInterval(timer1)方法时, 就是传入定时器 ID 进行停止的。

# 错误回顾

## 警告：Emitted value instead of an instance of Error



解决方法：

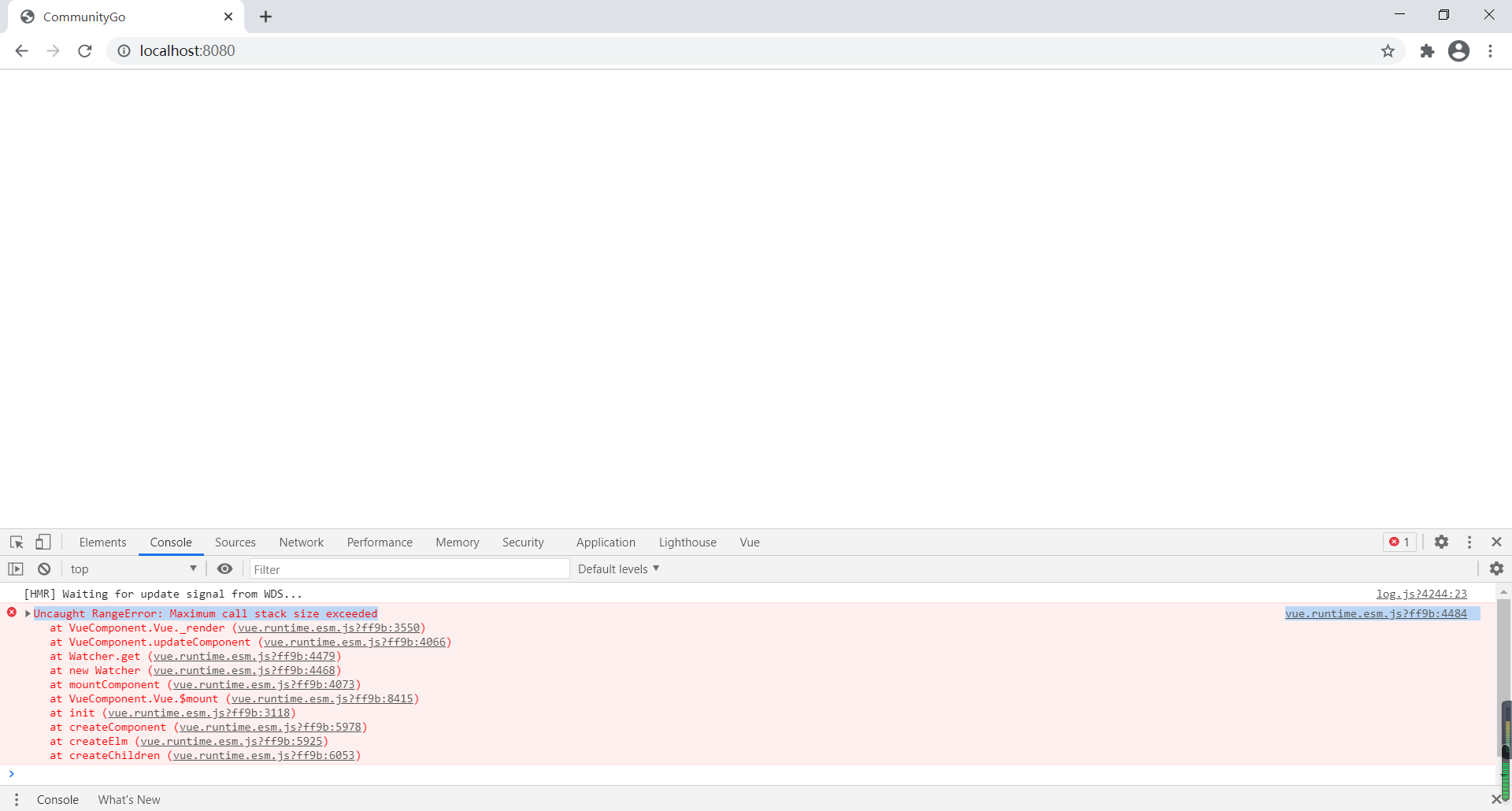


当使用v-for进行列表渲染时，为元素绑定一个key,即v-bind:key="yourKey"

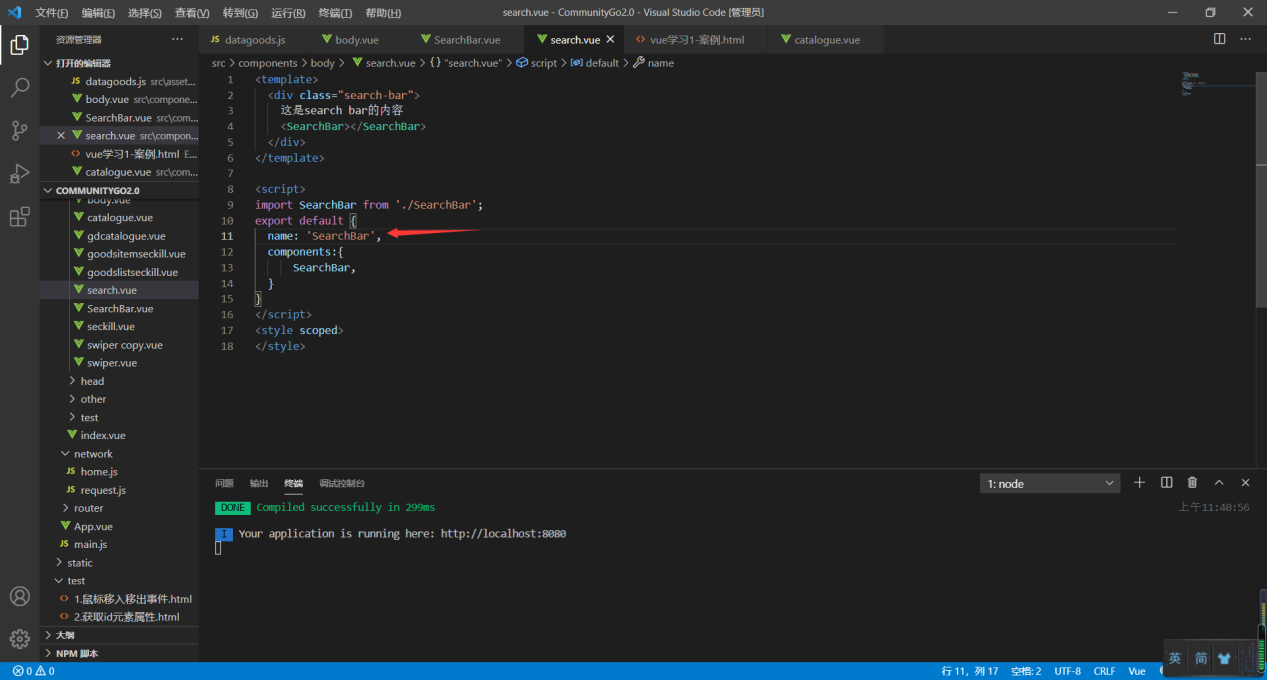
此种做法也是vue官方推荐的

v-bind:key="item.id"

## Maximum call stack size exceeded堆栈溢出

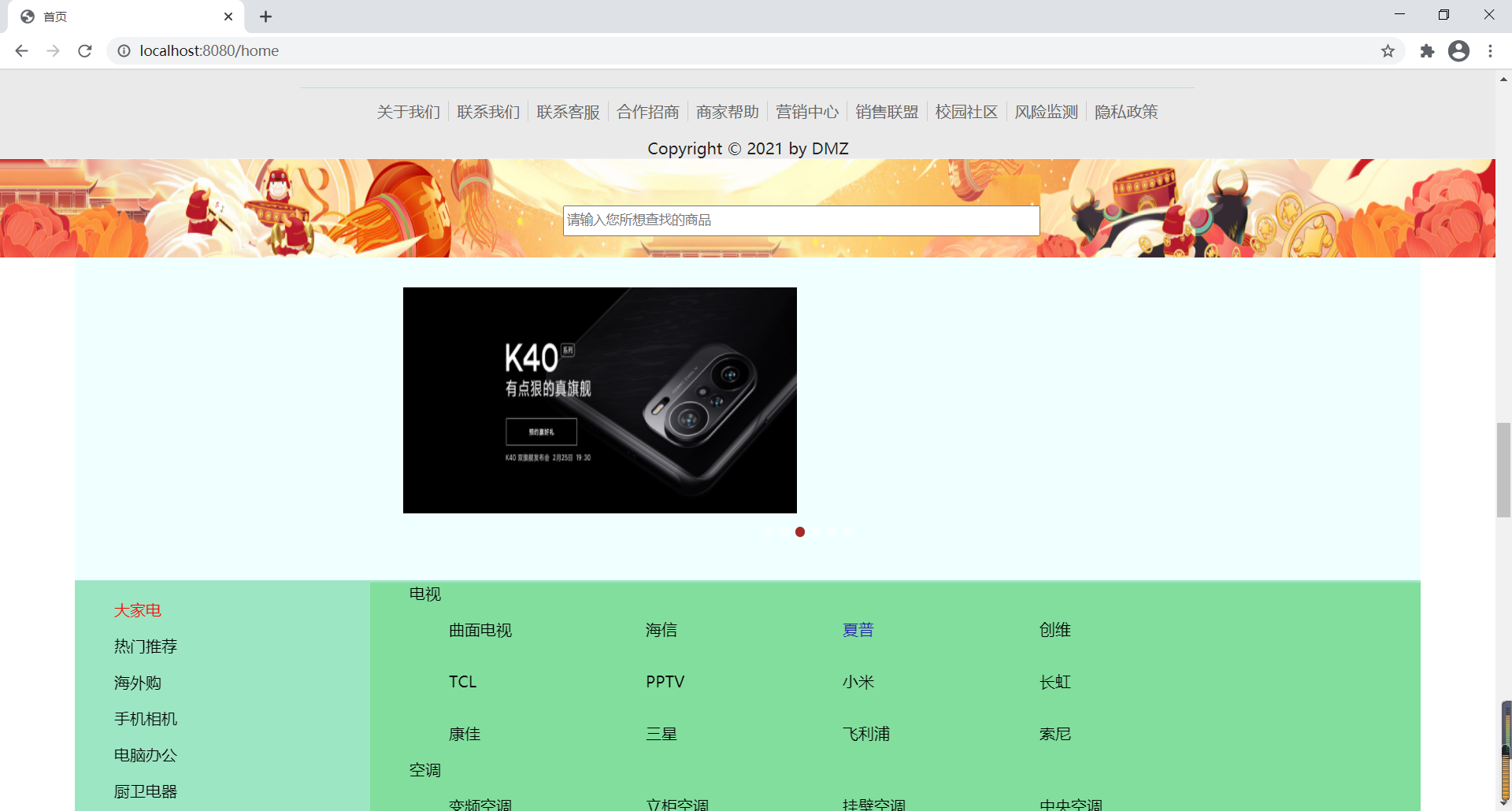


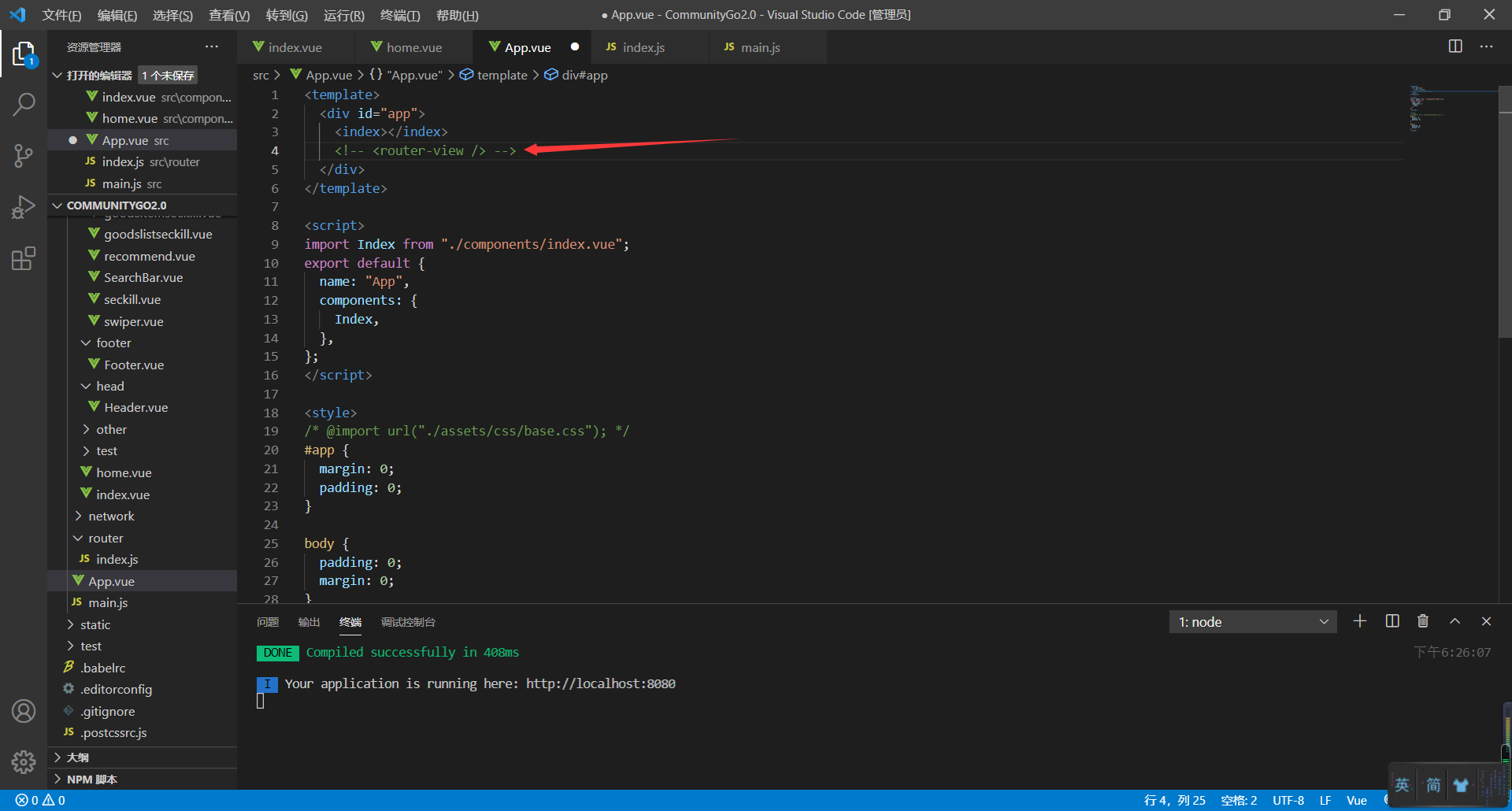
原本写法：



修改name即可，大写变小写

## 配置路由后页面布局错乱





App组件中渲染了两遍