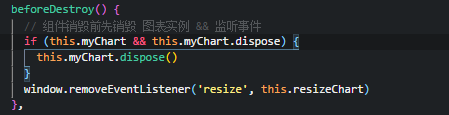
**关于1903-连锁组件中所遇棘手问题的解决方案**

**问题一**

**问题说明：**页面中使用tab切换echarts图表，会出现内存泄漏问题，导致浏览器内存过高而出现页面卡顿无法操作的情况

**问题分析：**页面中有多种图表，包括折线图、柱状图等，若使用v-if来切换图表的显示隐藏时，图表实例依然存在，没有被销毁； 且这些图表都带有监听事件，监听窗口大小，来实时改变图表的大小，切换图表时，这些监听事件也没有及时清除；因此会导致浏览器内存无法释放，从而倒是内存泄漏，操作频繁时会导致浏览器崩溃。（尤其在ie下）

**解决方案：**在图表组件中，将图表和监听事件在beforeDestroy 中做销毁清除处理



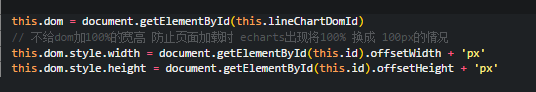
**结果：**内存得已控制

**问题二**

**问题说明：**echarts图表在页面加载完成后，放置一段时间，偶现图表被压缩成一团，图表容器变成100px\*100px的情况。（注：图表切换时没有用v-show，用的是v-if，切换时也没出现图表压缩的情况）

**问题分析：**可能是偶尔出现echarts的js获取不到容器的实际宽高，且设置了实际图表宽高为100%，echarts获取到的宽高为100px； 测试，当实际宽高设置为200%时，显示为200px。

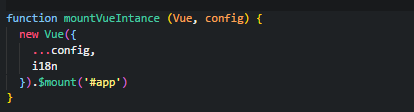
**解决方案：**图表容器的宽高不在less中设置成100%，而是实时获取其父级元素的宽高并赋值，可以避免此类问题。



**问题三**

**问题说明：**在除vue外js文件中的，语言包变量该如何获取

**问题分析：**关于多语言变量的获取，在一般的vue文件中，可以用$t(“xxx.xx”)来获取，在vue文件中的js，例如methods、或mounted等钩子中可以通过this.$t(“xxx.xx”)来获取，但若是在vue外的js文件中（例如一些js配置文件），则无法根据上述方法获取，因为i18n被挂载在vue根实例上。举例如下：



**解决方案：**可以把i18n绑定到window上，这样可方便vue以外的文件使用，获取相应的语言包变量。如下：



在除vue以外的js文件中的引用方式，如下：



**问题四**

**问题说明：**想要给某个对象新增多个属性并赋值时，会出现该对象监听不到该变化，且视图也不回更新。如下，因为根据选择不同时间维度，每次获得的时间变量不同，因此每次都要给对象requestParams新增不同的时间变量，但是在监听requestParams时，却监听不到变化。



**问题分析：** 因为vue.js在初始化实例的时候，会把data对象上的属性转为getter/setter，这样才使得属性变化能被检测到；而对于没有在data中定义的属性，直接新增属性并赋值，这些属性没有带上getter/setter这类属性方法，因此vue是检测不到其变化的

**解决方案：**可以通过$set()方法解决，this.$set(obj,key,value)



**补充：** 除对象外，想改变数组中某个索引对应的值，同理也可用$set()方法，例如，this.$set(arr,index,value)

**问题五**

**问题说明：**

**问题分析：**

**解决方案：**