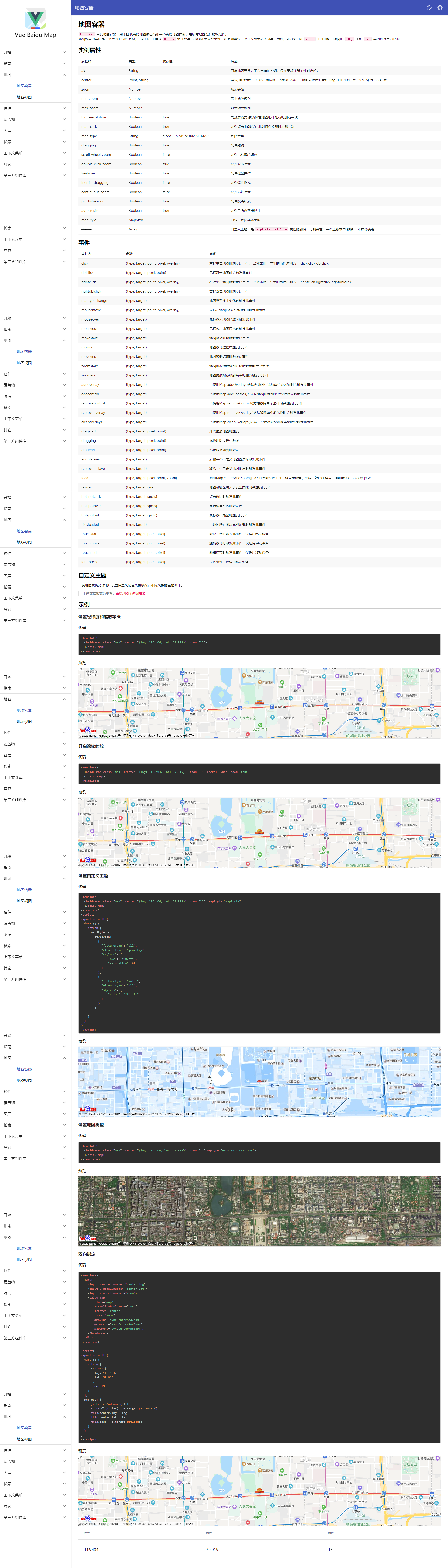
let ids = selectedData.map(item => item.id).join(",")



前置摄像头的地址：

192.168.1.250

后悔摄像头的地址是：

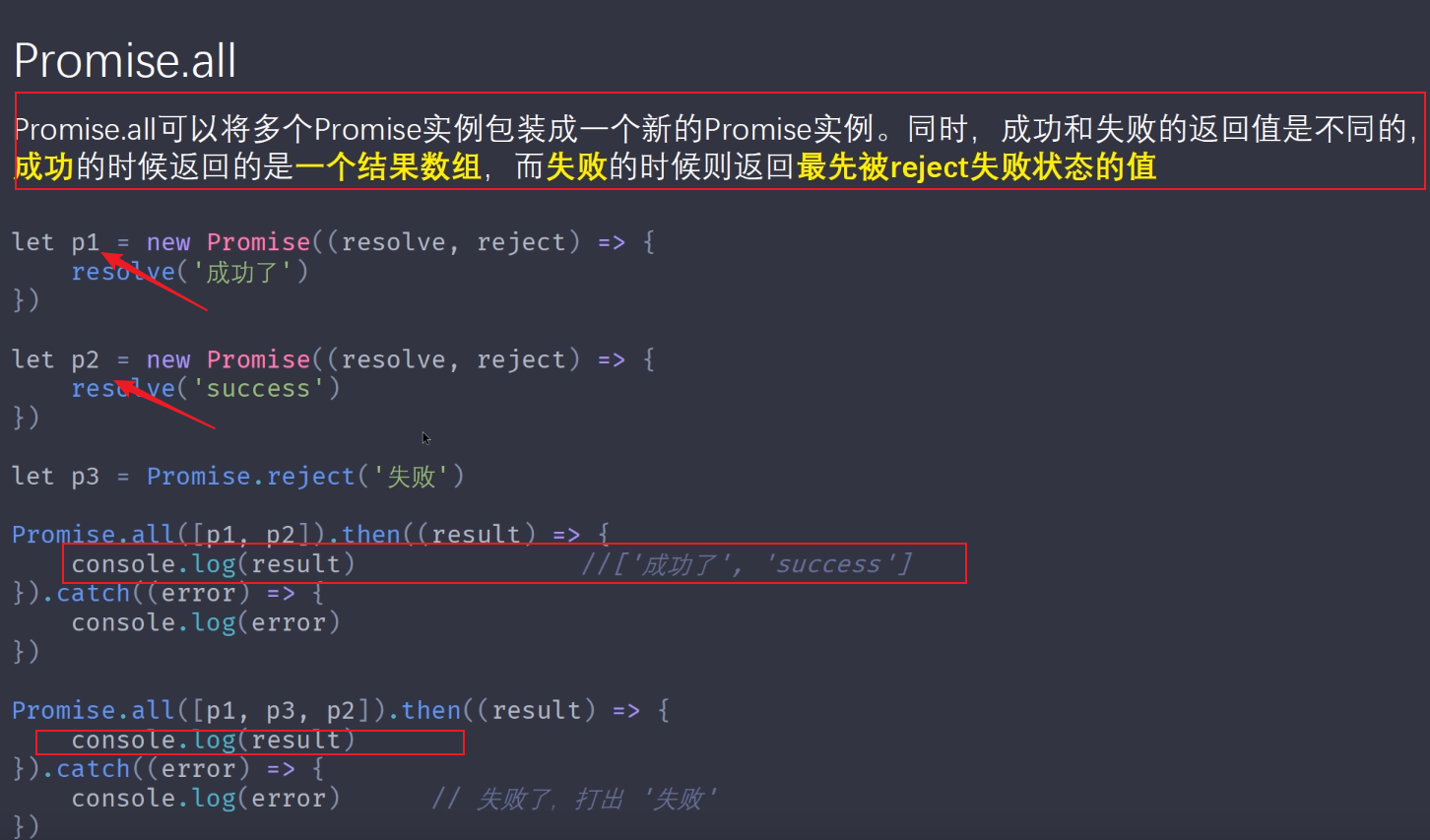
192.168.1.251

端口号启用的是：554



# JQuery篇

1. $.type() 函数用于确定JavaScript内置对象的类型，并返回小写形式的类型名称。
2. Promise

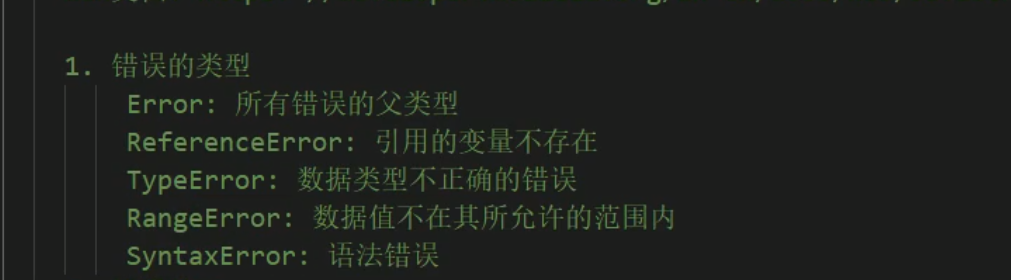


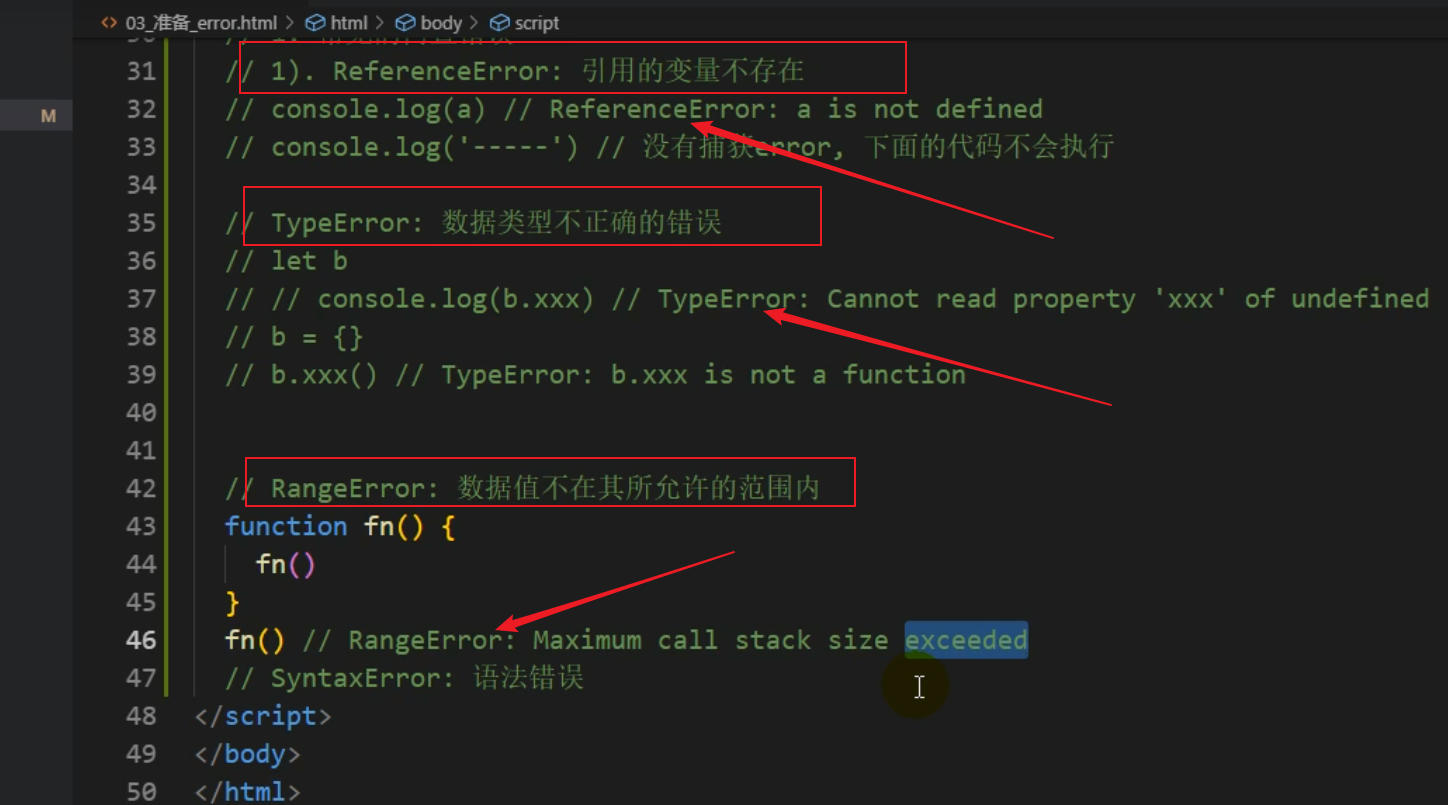


讲一个函数作为对象使用的时候就可以称之为***函数对象***

***Call() ,apply() ,bind() 只有函数对象才能够欧调用***

***常见的错误类型***







常见的错误处理：

A:捕获错误，B:抛出错误



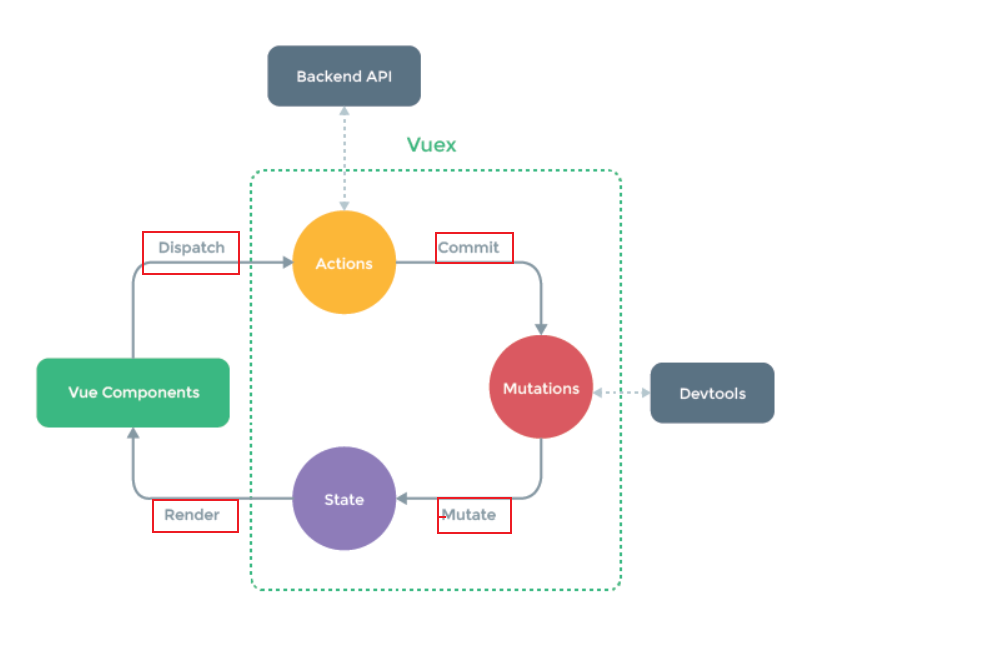
***VUEX：共享数据***

Vuex 中的核心功能包括是state , Mutation , Action

State: 用来声明数据（数据是动态的🡺响应式的，按照vuejs的规范state中声明的数据是不能够直接修改的，必须通过Mutation来修改）

Mutation：用来修改state中的数据，（同步执行），通过mutaion 修改的数据可以在开发者工具Devtools 查看修改的记录，如果不遵照vuejs的规范，不在Muation 中修改了state 中的数据，是无法在开发者工具中查看修改记录的，

Action: 用来异步获取数据（后台的数据），把获取到的数据通过Comnit 提交给Mutation进行修改，



showOverflowTooltip:false //表格中的列字段超出的部分不隐藏

**接口400:** 前端提交数据的字段名称或者是字段类型和后台的实体类不一致 或 前端提交的参数跟后台需要的参数个数不一致，导致无法封装；(前后端字段不一致，或者类型不一致)  
getFormModel( )获取表单中的信息

v-clock 解决因为加载而产生的的闪烁问题，加到标签上后会使其加载不闪烁

当前右侧layout面板，高度:🡺this.$store.digitalPark.contentHeight

租户用电数据中有弹框的新功能

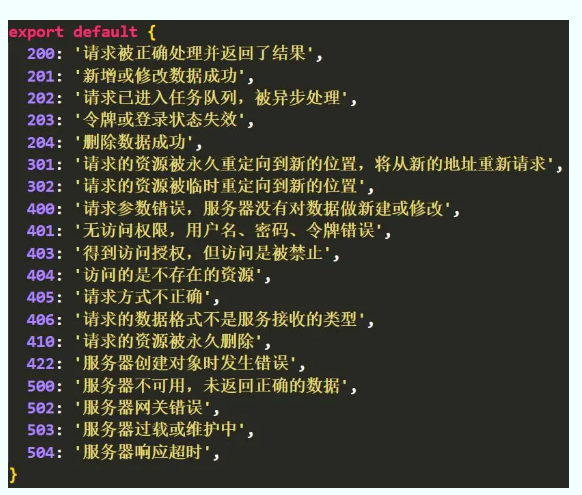
表格设置列标题和标题内容文字对齐方式

headeralign: 'rigth', align: 'right'



**返回状态码含义：**

**101：指定用户被拦截，==》关掉杀毒软件，或者是关闭防火墙等**



# Promise原型方法then, catch, finally

1.then() 方法可以处理 Promise 被接受或拒绝的情况。

2.catch() 方法用来处理 Promise 被拒绝的情况，相当于 then(undefined, onRejected)

3.Finally

A）.finally 方法用来指定在 promise 结束时，无论结果是 fulfilled 或者是 rejected，都会执行的回调函数。这样可以避免同样的语句需要在 then() 和 catch() 中都要写一次的情况

B).finally 方法的回调函数不接受任何参数，这意味着没有办法知道前面的 Promise 状态到底是 fulfilled 还是 rejected。这表明，finally 仅用于无论最终结果如何都要执行的情况，而不能依赖 Promise 执行结果

C）.finally 方法本身无异常抛出的情况下，总是会返回原来的 Promise 对象值；若抛出异常，则返回异常的 Promise 对象。



TypeScript学习教程：

TypeScript 是js的超集，它可以编译成纯 JavaScript。编译出来的 JavaScript 可以运行在任何浏览器上。TypeScript 编译工具可以运行在任何服务器和任何系统上。TypeScript 是开源的。

# 前端工程化的理解

截止到现在，从事前端开发已经两年左右，对前端工程化有着在工作积累中的理解，首先来说这是一个比较大的概念，我觉得到现在为止，没有人能够给他下一个完整的定义，无数前端开发工程师，对前端功能化都有着自己独到的理解和概括，这也是迄今为止为啥没有一个确切的定义来描述这个词语，如果让我来说的话，我个人理解前端工程化就是**凡是能够提升开发效率，提高前端开发质量的手段和工具都可以理解为前端工程化**

从提升开发效率来说：

举一个简单的例子：原生js 中的单击事件绑定，通过dc.get…onclick = fun …,到Jquery中的$().click(function() { }),再到vue中@click等于某个事件，代码的简洁度在逐渐提高，同样是一件事，实现功能的代码变少了，可读性变的更强了，所以开发效率提高伴随着代码简洁度的提高，另一方面也直接反映出如果开发效率要想提高，开发框架技术也要得到发展，从原生js,Jquery，再到vue一步步得到扩展

从降低大型项目的开发难度来说的话，一个大的项目被拆分成若干个小的模块，难度被降解，模块化和组件化在vue项目中被体现的很明显

设置视口

<meta name=’viewport’, content=’width=divice-width,user-scalable=no, initial-scale = 1.0’ >