## 面经项目 - 接口地址-接口文档地址：

[https://www.apifox.cn/apidoc/project-934563/api-20384513](https://www.apifox.cn/apidoc/project-934563/api-20384513)

接口基地址：[http://interview-api-t.itheima.net/](http://interview-api-t.itheima.net/)

## vant介绍和基本使用

- 官网：

[https://vant-contrib.gitee.io/vant/v2/#/zh-CN/](https://vant-contrib.gitee.io/vant/v2/#/zh-CN/)

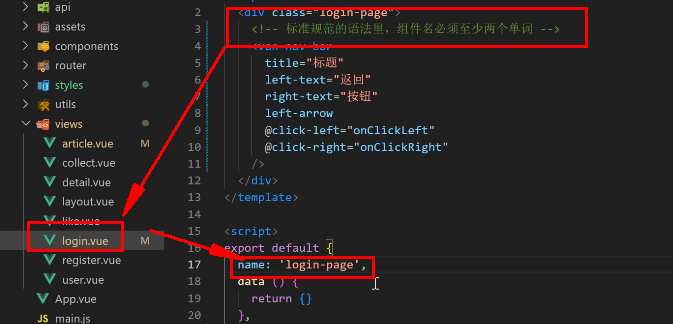
- 演示地址：[http://xpzll.gitee.io/hmmj-demo](http://xpzll.gitee.io/hmmj-demo)

- 账号建议自己注册，每个人用自己账号避免冲突

步骤：起码知道这是啥东西呀。

1. 搞项目配置技术栈。，然后设置路由，设置主题
2. 登录页布局，导航栏和表单
3. 注册页布局
4. axios封装，注册接口，完成注册

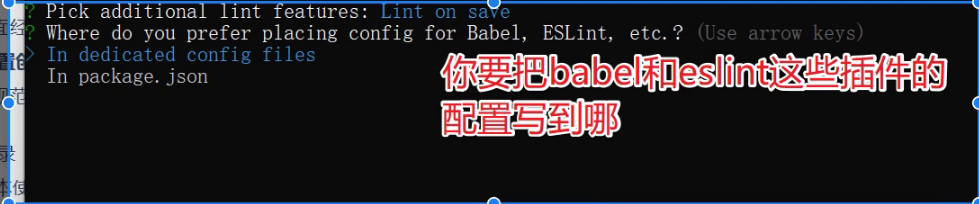
注意：标准规范语法里，组件名必须至少两个单词。name可以改成大驼峰。为什么文件名可以是单个单词就可以了，是因为加了name属性。



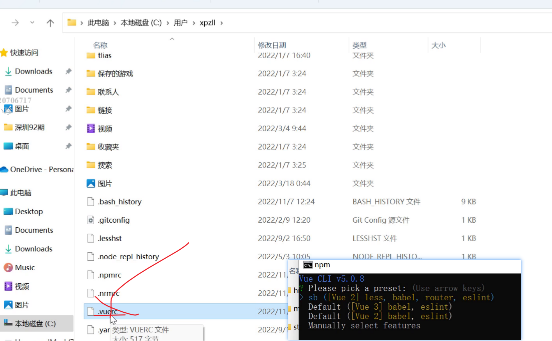
直接在演示地址注册一个。

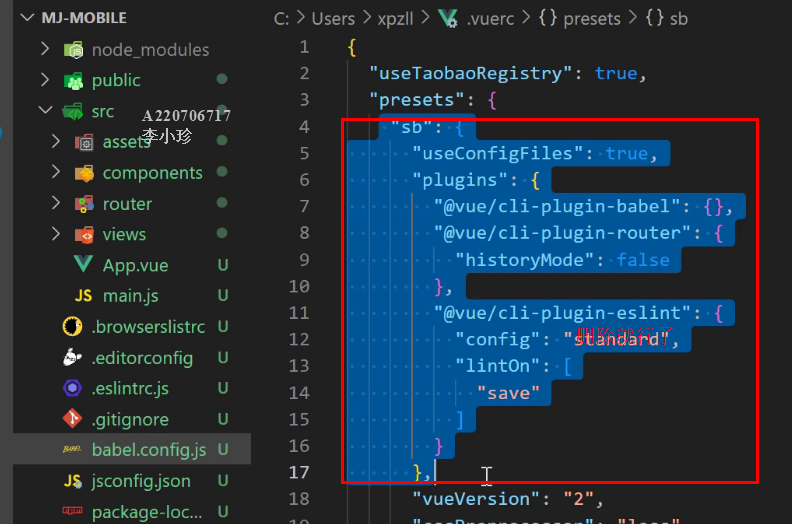
不管项目多么复杂都需要把大概的架子搭起来。Vue creat 项目名字

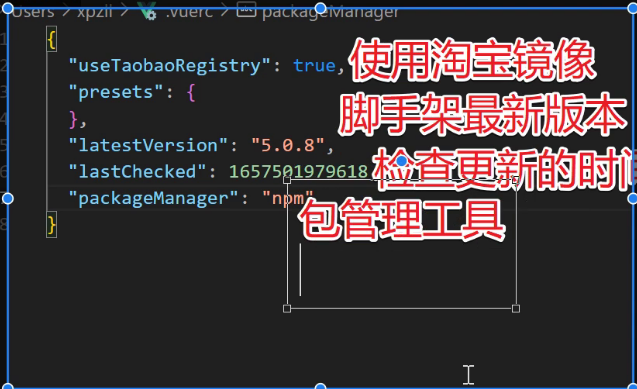
之前选择默认配置没有路由，默认配置有babe(es6降级为es5)，Vue2，eslint。所以后面需要自定义设置配置。



如何删除保存的配置？



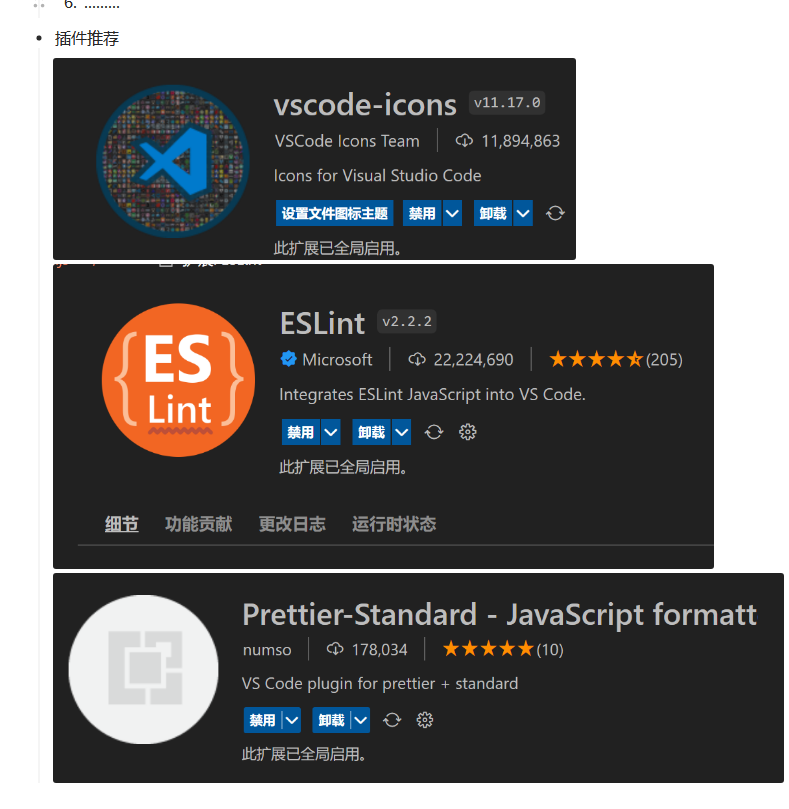




nrm可以管理npm的下载源。npm config set registry .----

nrm会帮你自动安装最新版本 npm i nrm -g

不要尽信百度很大有可能过时了。



放组件库-放组件的仓库。

Vant移动端组件库，也可以移动web和微信小程序。Vant是2的版本,vant3和vant4是为了Vue3准备的。组件库：提供大量帮我们组成网页的一部分，要什么就拿什么。

vant的使用看官网。注意版本别用错了。



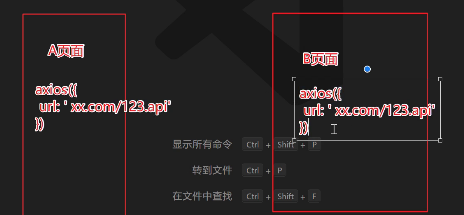
采用的是全局导入。

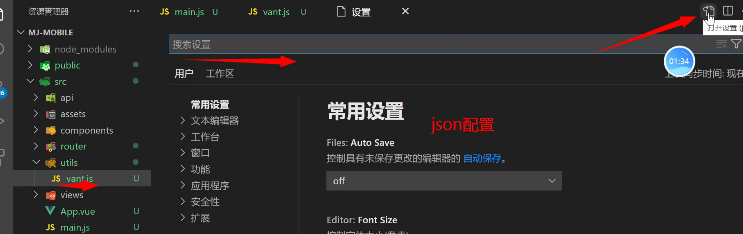
文件夹说明：api和utils。发请求，就不要直接发请求，而是封装成一个函数，好处就是A页面，B页面，都要对一个地址XXXCOM/123.API接口发请求，两个页面都写相同请求会造成代码冗余，另外网址发生改变，两个页面接口都要改，耗时麻烦不利于维护。同一个代码多个地方要用，封装成一个函数。A页面要用,B页面也要用。所以把他封装到一个独立文件夹API文件夹里面进行导出，然后该相应页面导入就行了。

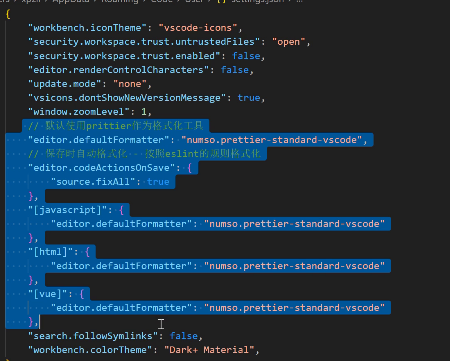
utils工具文件夹放项目要用到的一些工具。封装的一些函数或者第三方插件的导入一些。

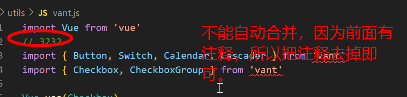
因为如果全部写到main.js里面代码越来越多，不利于阅读。、

所以在main.js直接导入utilshe和api。Import ‘@util/vant’









插件没那么智能。

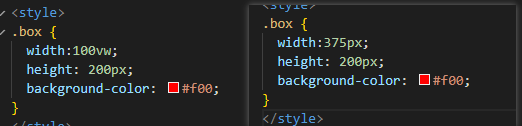
vant有3种方式：手动导入耗时，全部导入体积大，推荐自动按需导入。看官网说明文档。

移动端VW适配方案。

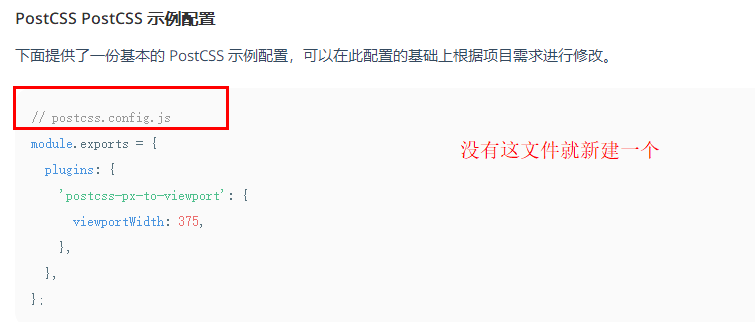
移动端特点：每个屏幕尺寸不一样。默认有个内外边距。移动端要解决的第一个问题就是屏幕适配。换成vw，即便是ipad也能适配。有个问题：美工给到的设计稿是px，比如是50px。自己做转换。比如设计稿屏幕是以375作为底稿，50/(375/100)=16.3vw比较麻烦。所以需要插件。另外考虑到万一发现布局有问题要来检查看到底是哪个样式出问题了。设计稿是px单位，用插件之后变成vw单位，所以不方便对设计稿排查问题。所以既要代码中还是写px。但是实际代码运行过程中他会自动转成vw，这样方便比对代码问题，也解决适配问题。

vant组件库官网文档也说明了一种进阶用法：浏览器适配，推荐使用postcss插件，将px单位转变为vw/vh单位。

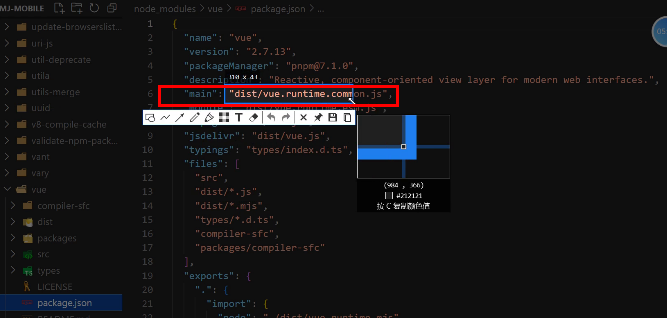


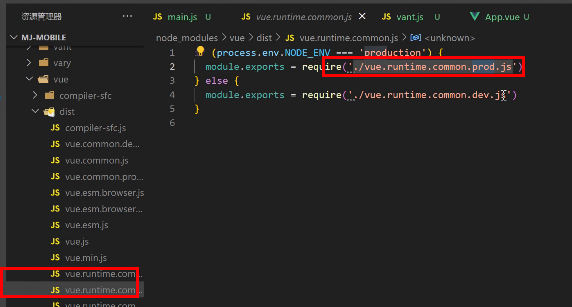
) 

任何插件下载都是npm i 插件名字。下载不行就加-f。



Import Vue from ‘App.vue’不可以，细节：在模块化管理项目里，脚手架,node,导入时不管你用require还是import导入，如果加入相对路径，那相对路径是什么就找什么文件。但是没有加任何相对路径，那么它其实就是找node\_modules里的文件夹里面的package.json然后再找到package.json里的main指定的文件，再把这个文件导入进来。导入vant





直接导入，只是希望把这个代码执行一遍。就直接写路径，不写form，另外没有暴露出去。

所以知道有没有加from，就看有没有暴露出去。 不暴露就不加。不管加不加都相当于把代码执行了一遍。



在前端里面，但凡导入一个文件，不管是什么方式导入，都相当于把这个文件内容当js代码执行一遍。





分析页面：一级路由和二级路由。

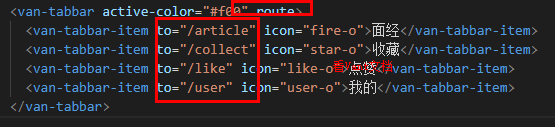
一级路由：没有包装如何。骨架。 二级是具体的内容。配置路由出口。

导入，如果是js文件，vue文件可以省略不写后缀

[layout](https://fanyi.so.com/" \l "layout" \t "https://www.so.com/_blank)：布局.vant的tabBar跟索引做绑定，实际项目我这边要的是跟路径做绑定。

之前webstorm很经典，臃肿，强大，由于收费，所以使用vscode

tabBar路由模式开启：看vant文件然后看到路由模式，不一定需要点击事件然后this.$route.push。vant的to属性是否开启路由模式。看文档



对于组件库的一些东西不需要背诵下来，原生JS需要记忆。

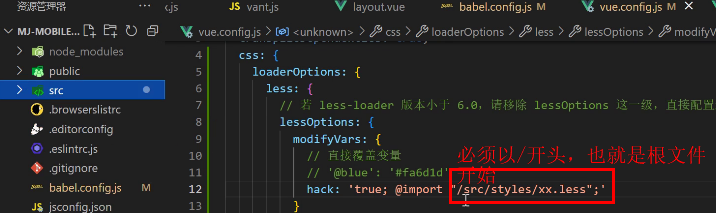
修改vant的主题：vant的东西就去找vant。主题这东西一般组件库里面有说明。

我们用的脚手架vue-cli搭建的项目，不是用webpack搭建的项目。

一个颜色多个地方要用，less里面可以声明变量。为了保持风格统一。

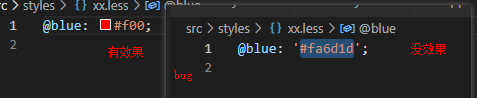
改了配置文件要重新启动才能有效

一般把less文件或者css文件都会放到styles文件里面



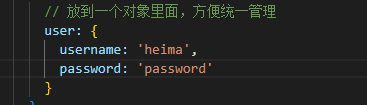
/代表最外面src里面的。。less要分号结尾，js不一定要分号结尾。

有个问题启动要6s。



在js文件和less文件语法不一样，所以注意下。

主题修改按官网重启运行，多次之后成功，不知道哪出问题。



label

<!-- 字符串 --><router-link to="/home">Home</router-link>相当于a链接

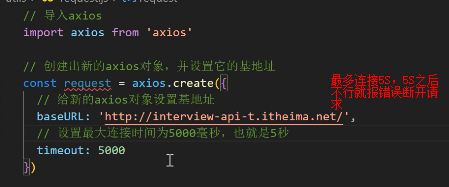
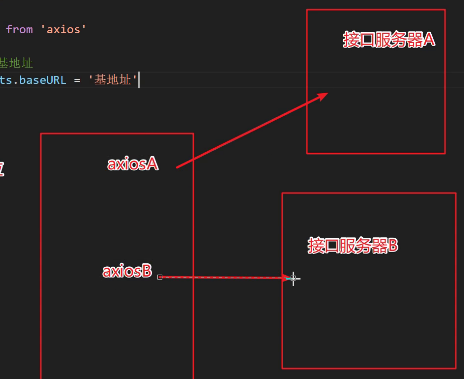
校验规则也看vant官网说明。

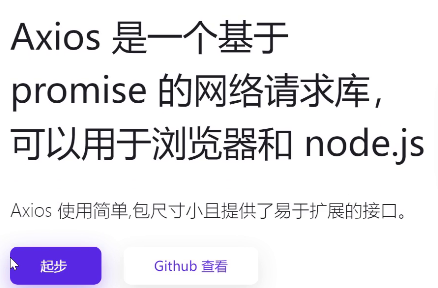
注册页：跟登录页面是一样的。复制粘贴一把梭。

发请求就需要下载axios。axios需要设置一个基地址。

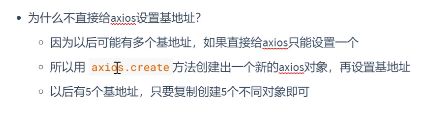
1. 导入axios2、设置axios基地址。2、设置请求拦截器和响应拦截器。

比如一个大型的项目它的接口放在不同的位置服务器上，所以有不同的接口基地址，所以需要创建不同的axios对象对应不同的基地址。axisoA，或者axiosB对应不同接口基地址发送到不同的服务器。所以就创建新的axios对象，并且设置它的基地址。

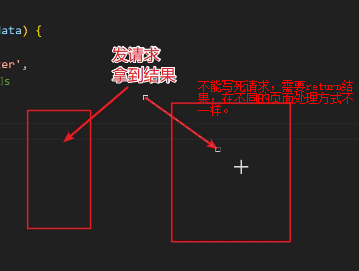


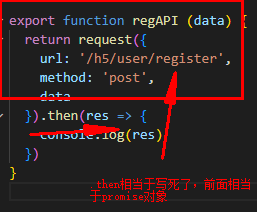


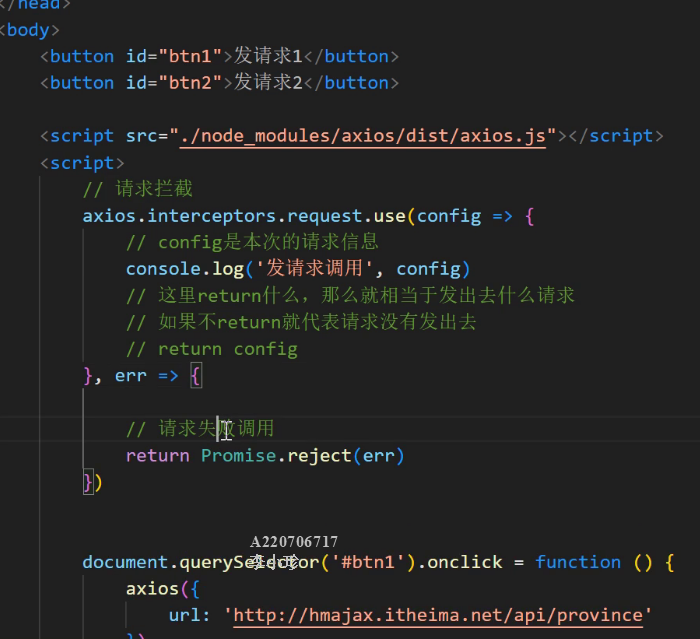
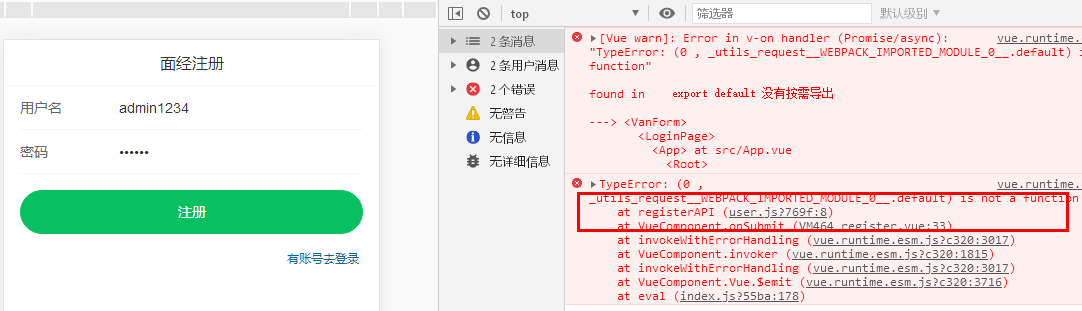
axios不会用就打开官网百度axios.，但是注意修改不同的单词，意思是一样的。

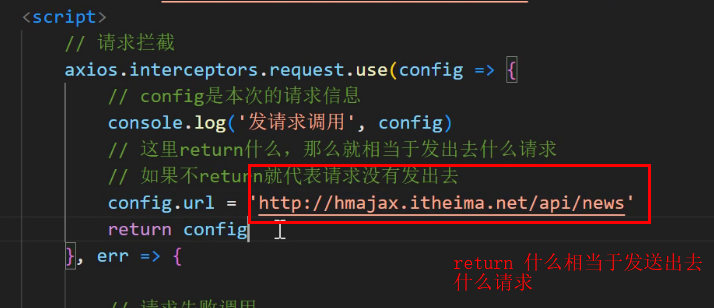


api文件夹封装接口，以模块名命名js。User.js专门发用户相关的接口，例如用户注册，用户登录，获取用户信息等。方便分门别类的管理。





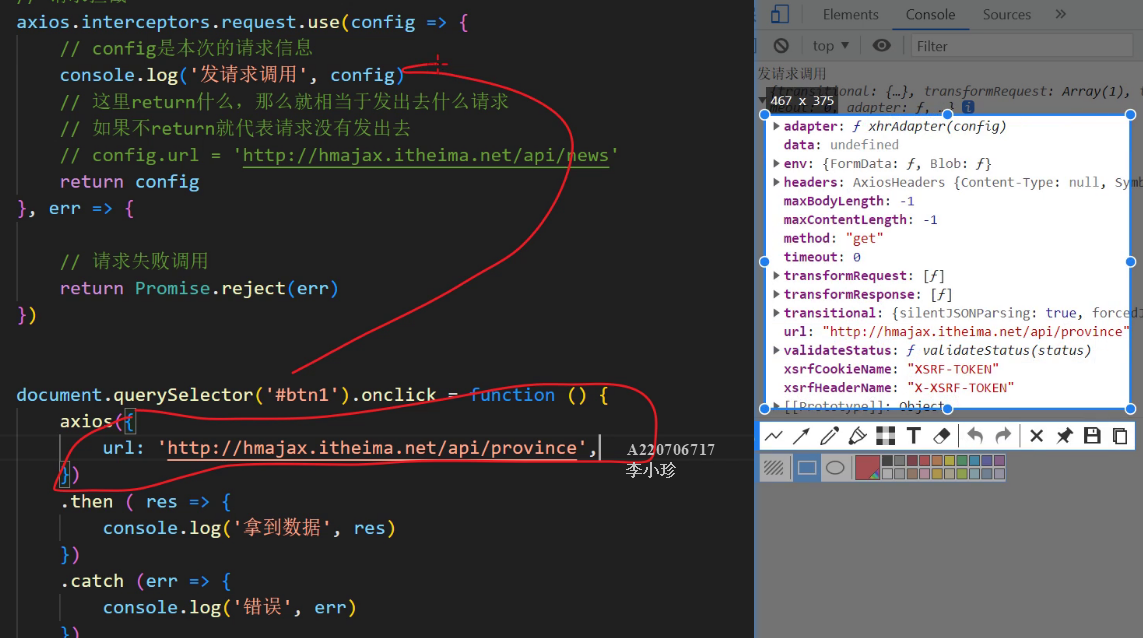




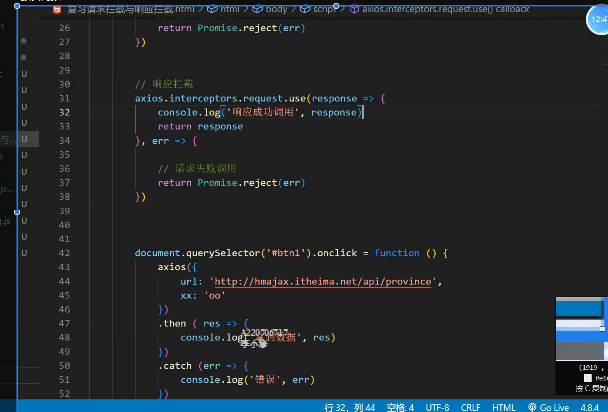
发送请求携带token。

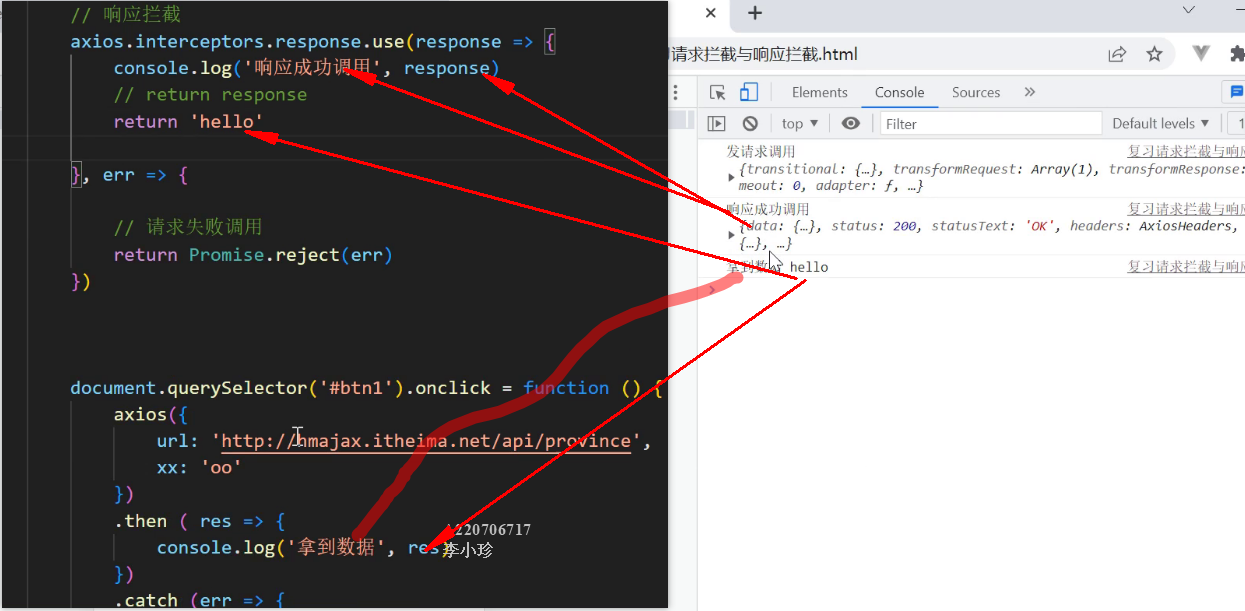




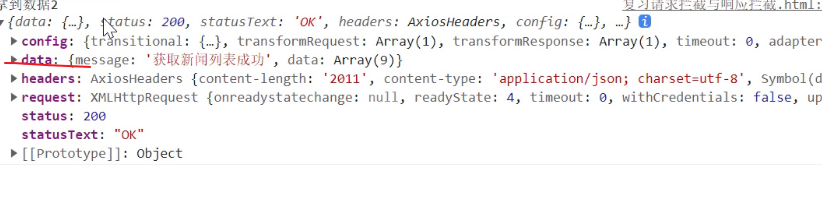


虽然只是写了一个，但是会给其它设置默认值。





响应拦截的作用：不管发什么请求，服务器返回数据都多包了一层，



所以在响应拦截直接加res.data

数据剥离：把data剥离出来，外面用的时候就可以少写一个data。原因：用axios发送请求，每次都会在外面多包一层，而我们不需要，所以在响应拦截直接剥离一下。