1. Axios

什么是axios？

官方解释：Axios 是一个基于 promise 的网络请求库，可以用于浏览器和 node.js，Axios 使用简单,包尺寸小且提供了易于扩展的接口，在客户端（浏览器）它使用 XMLHttpRequests。

1. 安装axios

npm install axios

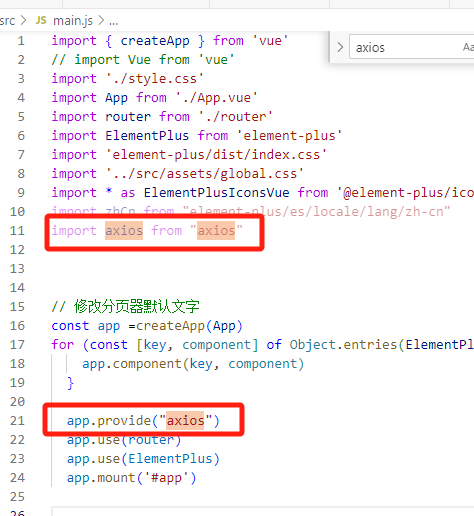
安装完之后可以在项目看到已经安装完成：



该axios的版本是1.6.8的。

1. 使用axios 以下代码也可以写在单独的html文件中
2. 导入该模块

在js部分写入：import axios from "axios" 引入之后就可以使用，这种引入是局部引入，当然也可以使用全局引入在main.js文件中写入：



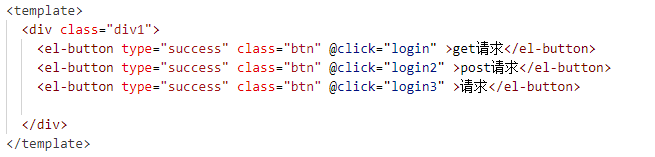
然后在其他地方引用时：



示例 拦截器

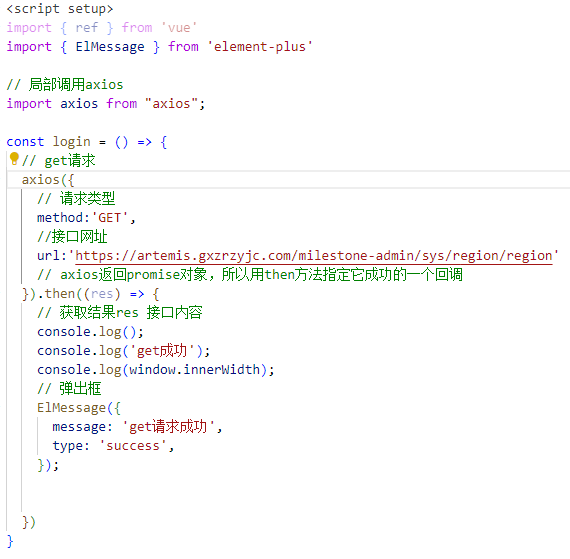
1. 发送get 请求

现在helloworld单位写几个按钮（引入组件），给按钮绑定对应的方法





现在实现axios的get请求，在helloworld文件的js部分引入axios ，然后在按钮的登录方法中，写入axios ,如下图：



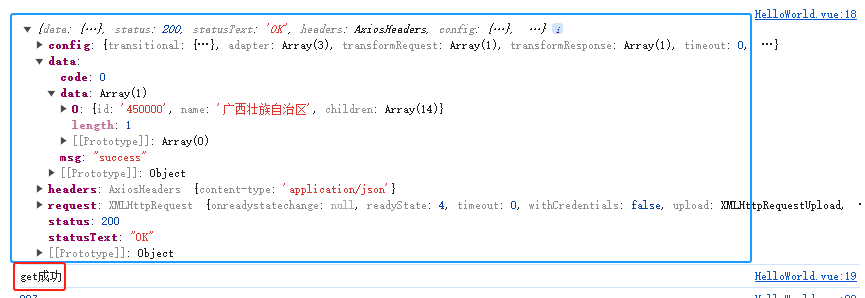
看页面结果：



点击get请求按钮，跳出消息，显示成功，



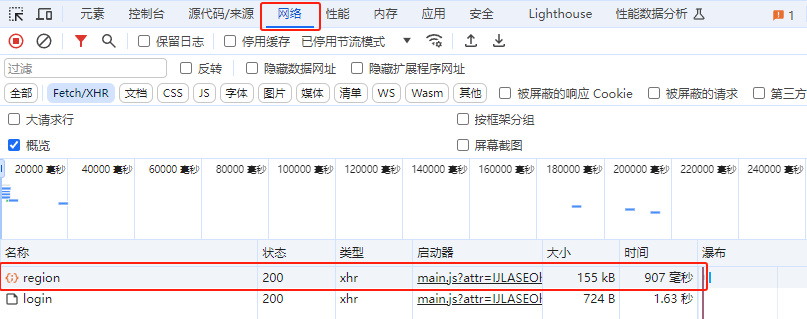
点开可以看到下图：



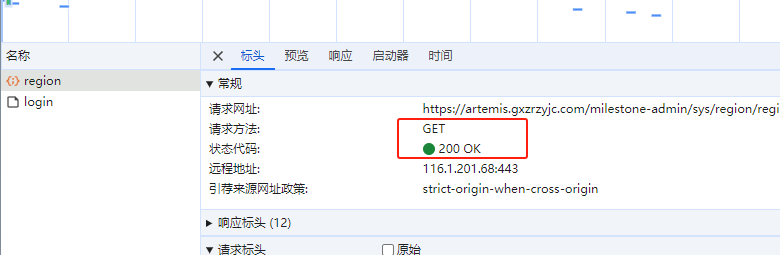
蓝色框是console.log(res);接口的内容的输出

红色框是 console.log('get成功')的输出

查看网络请求的状态



显示200 ，证明请求成功了，点击region进去可以看到



点击响应，可以看到接口的数据

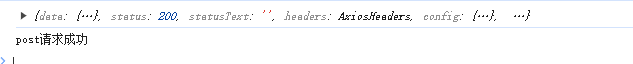


Post请求：

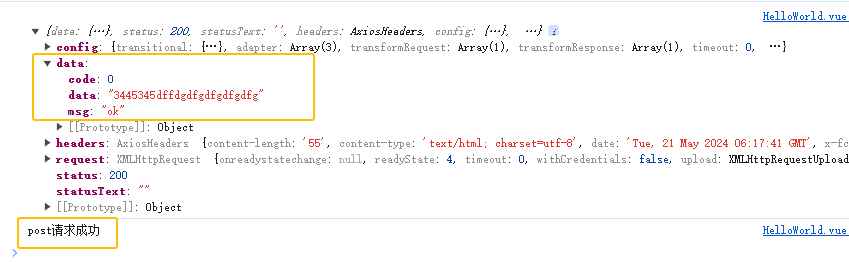
Js部分：



查看结果：

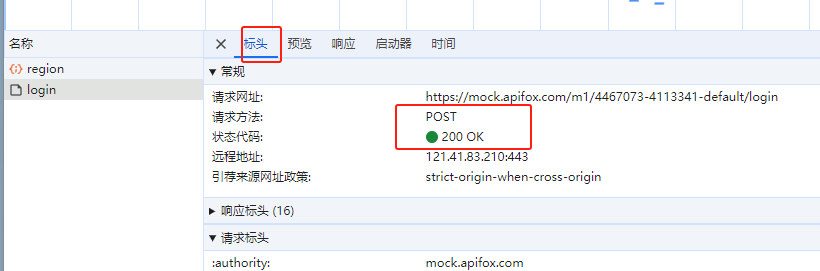


打开查看

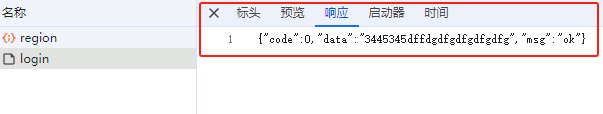




点击login查看一下：



查看接口消息，如下图：



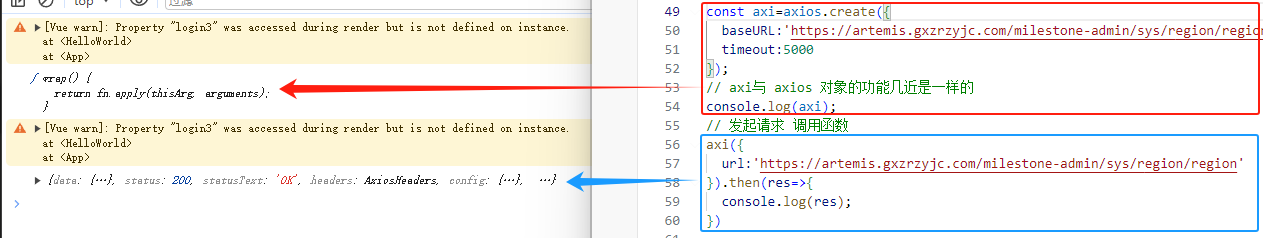
出现这些情况说明请求成功

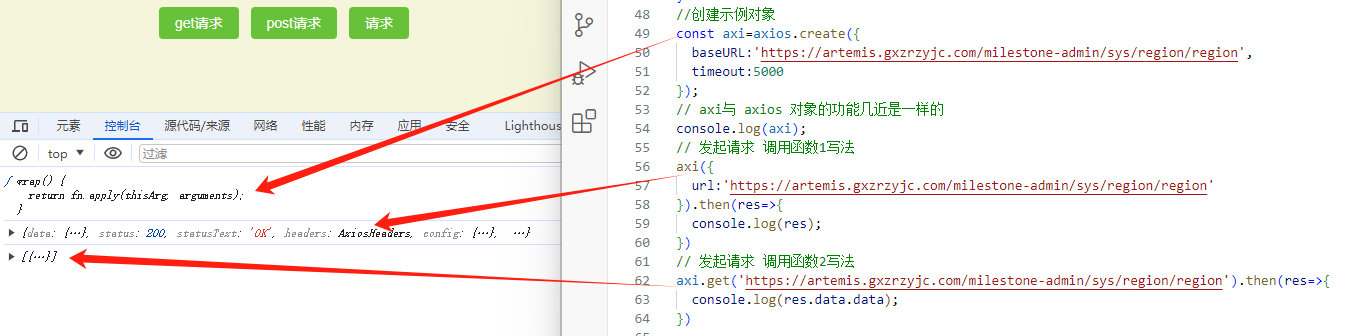
1. 创建示例对象

在js部分写入以下代码：



然后控制台输出以下：





1. 拦截器

拦截器其实是一种函数，分为两大类，一个是请求拦截器，一个是响应拦截器，



控制台输出结果：

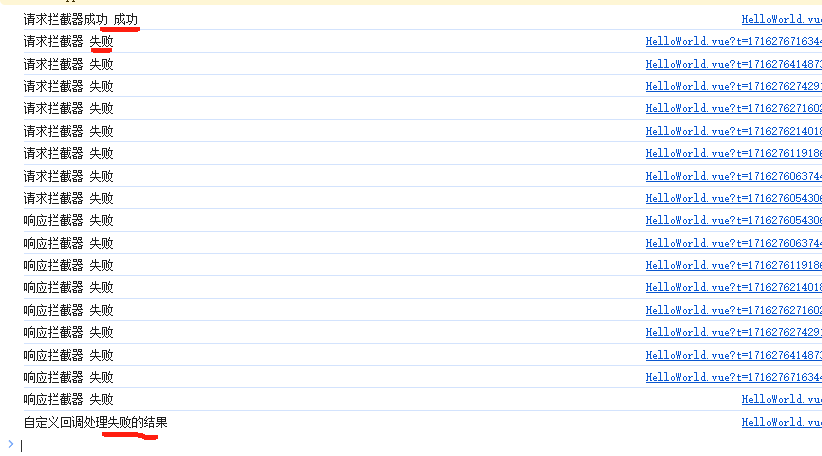


这个代码是先走请求拦截器成功 成功的回调，然后走响应拦截器 成功的回调，然后是自定义回调处理成功的结果成功的回调

如果是失败的话的：

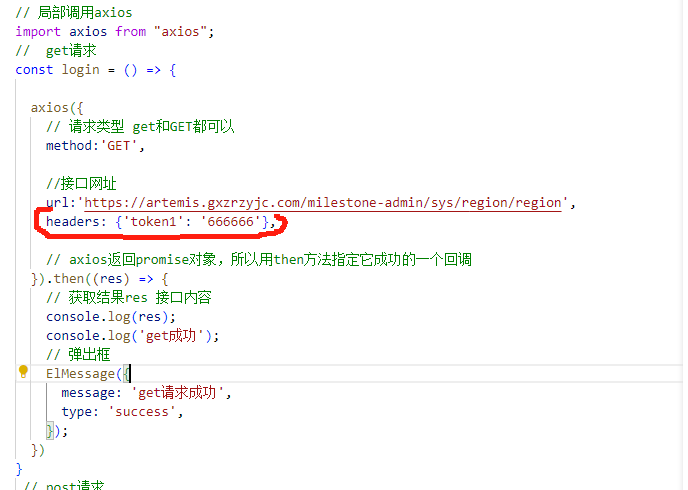


输出结果是：

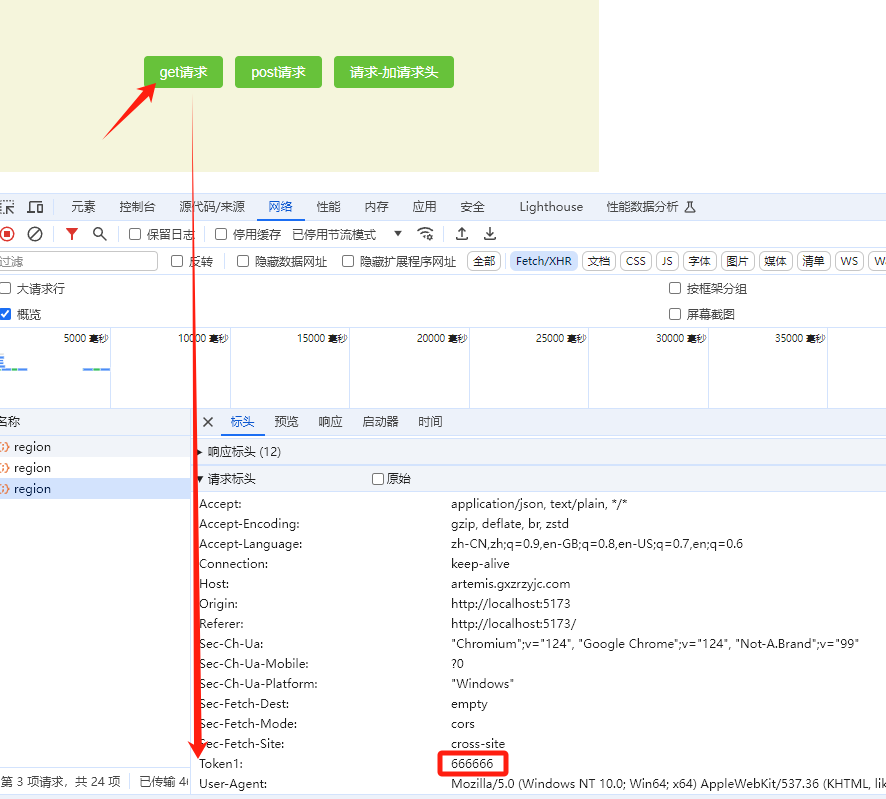


和promise很像

⑥请求头的添加

Get post的请求添加自定义请求头参数：

定义了请求头叫做666666，查看输出结果：

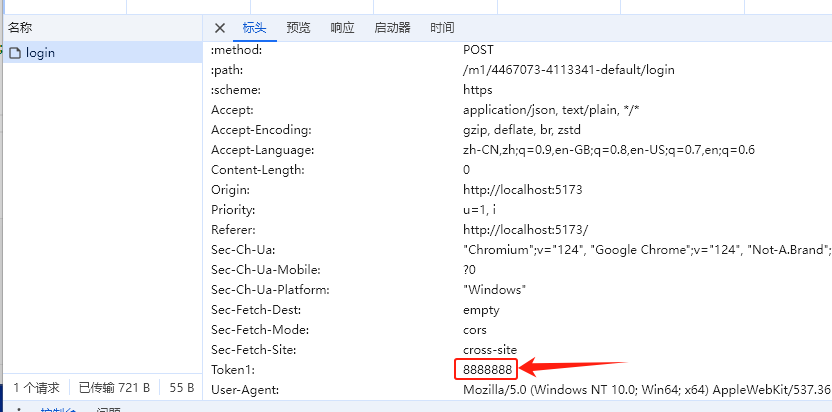


可以看到请求标头的信息是666666。

Post方法的请求头，这次获取的是接口的token消息，加到请求头之中：

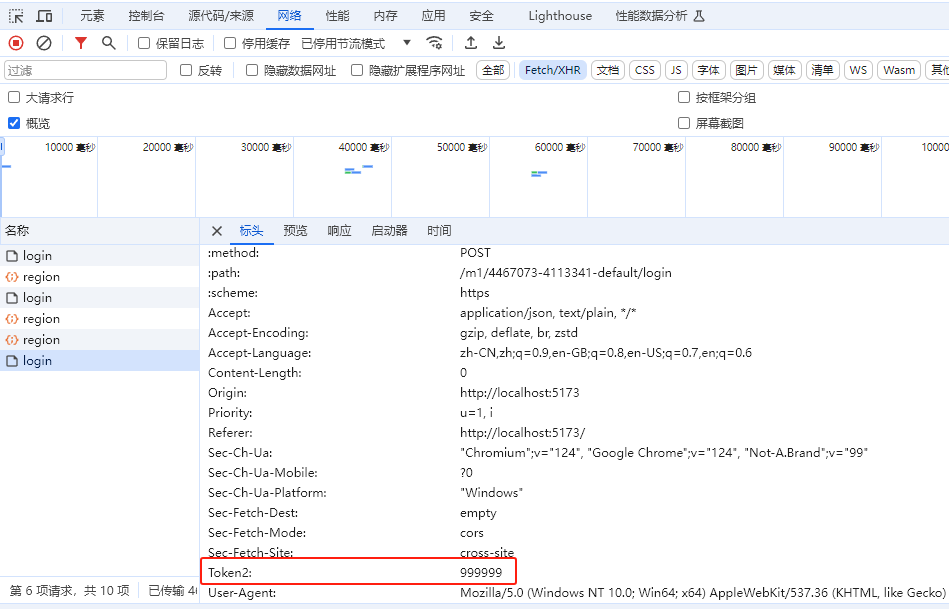


888888存在请求头中：

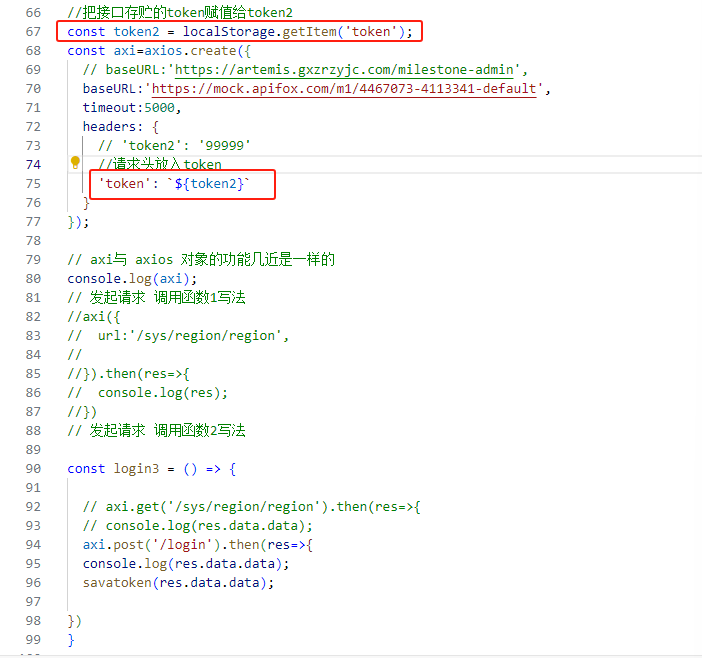


Axios示例中加请求头：

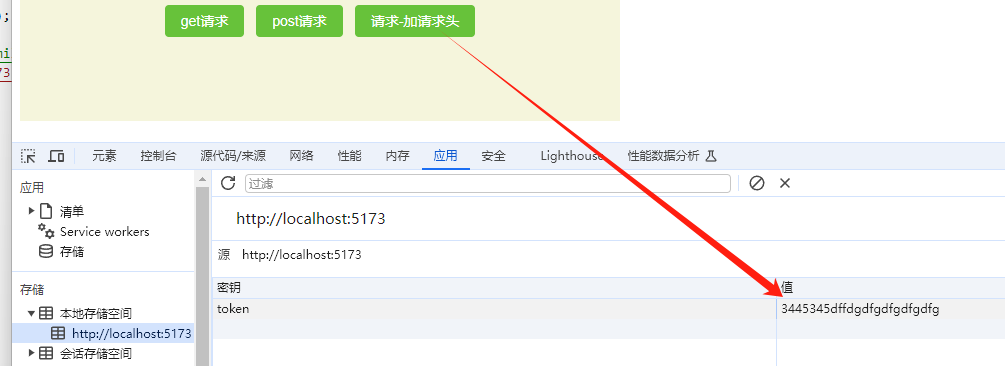




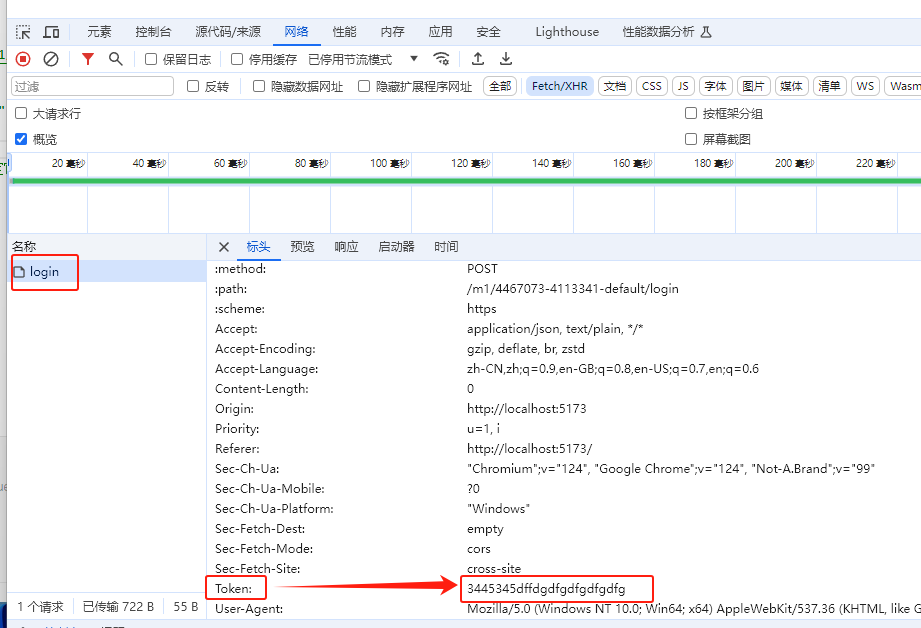
把接口的token放入请求头：



存贮的Token是3445345dffdgdfgdfgdfgdfg



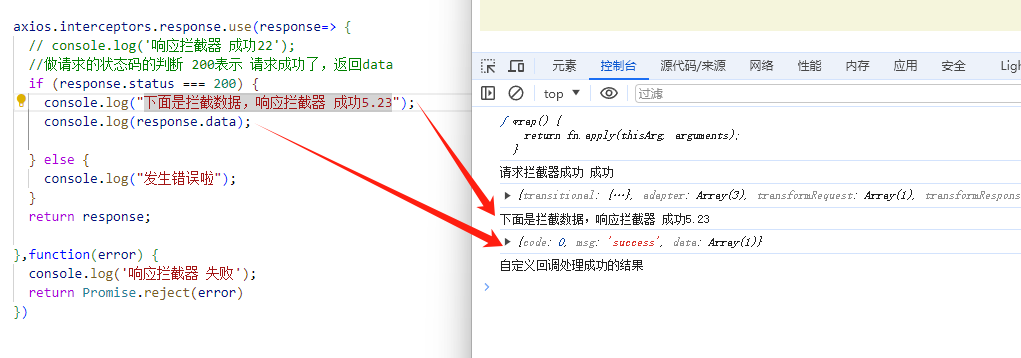
请求头带上了token:

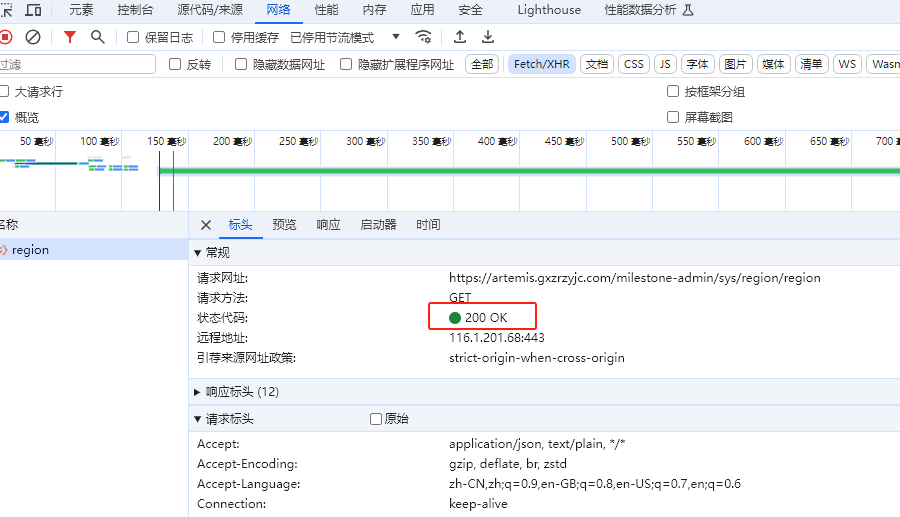


⑦ 拦截器给请求头带上token：



⑧ 响应拦截器做判断：





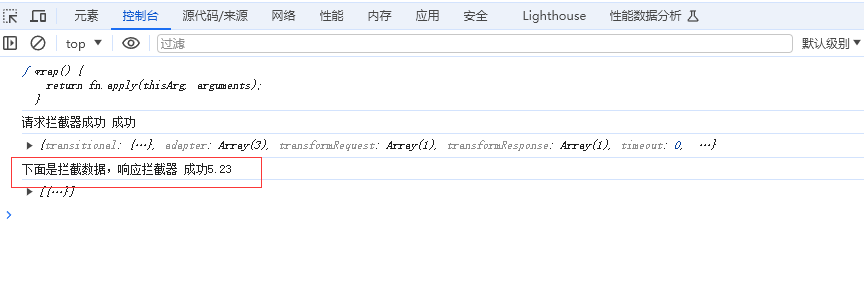
这个请求是200，执行条件为真的语句，所以是成功返回了接口的data数据

拦截器如果接口数据中的code=0 就成功：



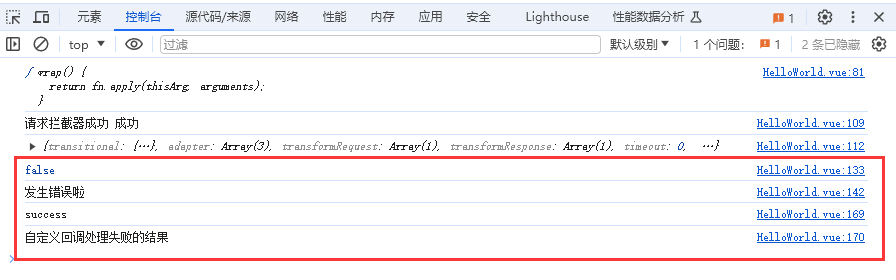
执行是拦截器对接口的code进行验证

输出的语句就是对的：



判断的code等于100，条件为false（因为接口code为0 ，等于100就是false)，就执行else里的语句：





当响应拦截器执行else的话，在axios的方法里res(回调数据)返回的是response.data.msg（success)的数据