Axios

1. Axios 是一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js 中。

2. // `params` 是即将与请求一起发送的 URL 参数

// 必须是一个无格式对象(plain object)或 URLSearchParams 对象

params: {

ID: 12345

},

3. // `auth` 表示应该使用 HTTP 基础验证，并提供凭据

// 这将设置一个 `Authorization` 头，覆写掉现有的任意使用 `headers` 设置的自定义 `Authorization`头

auth: {

username: 'janedoe',

password: 's00pers3cret'

},、

4. // `validateStatus` 定义对于给定的HTTP 响应状态码是 resolve 或 reject promise 。如果 `validateStatus` 返回 `true` (或者设置为 `null` 或 `undefined`)，promise 将被 resolve; 否则，promise 将被 rejecte

validateStatus: function (status) {

return status >= 200 && status < 300; // 默认的

},

5. // `httpAgent` 和 `httpsAgent` 分别在 node.js 中用于定义在执行 http 和 https 时使用的自定义代理。允许像这样配置选项：

// `keepAlive` 默认没有启用

httpAgent: new http.Agent({ keepAlive: true }),

httpsAgent: new https.Agent({ keepAlive: true }),

// 'proxy' 定义代理服务器的主机名称和端口

// `auth` 表示 HTTP 基础验证应当用于连接代理，并提供凭据

// 这将会设置一个 `Proxy-Authorization` 头，覆写掉已有的通过使用 `header` 设置的自定义 `Proxy-Authorization` 头。

proxy: {

host: '127.0.0.1',

port: 9000,

auth: : {

username: 'mikeymike',

password: 'rapunz3l'

}

},

6. // `headers` 是即将被发送的自定义请求头：

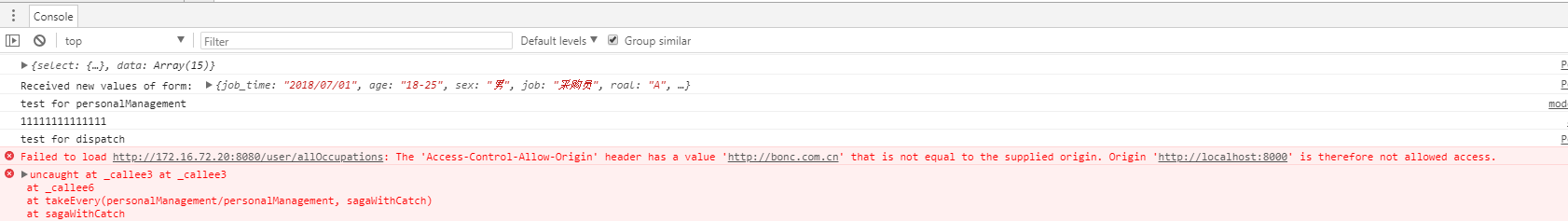
headers: {'X-Requested-With': 'XMLHttpRequest'},

可见headers的属性有很多个

7.一个例子：

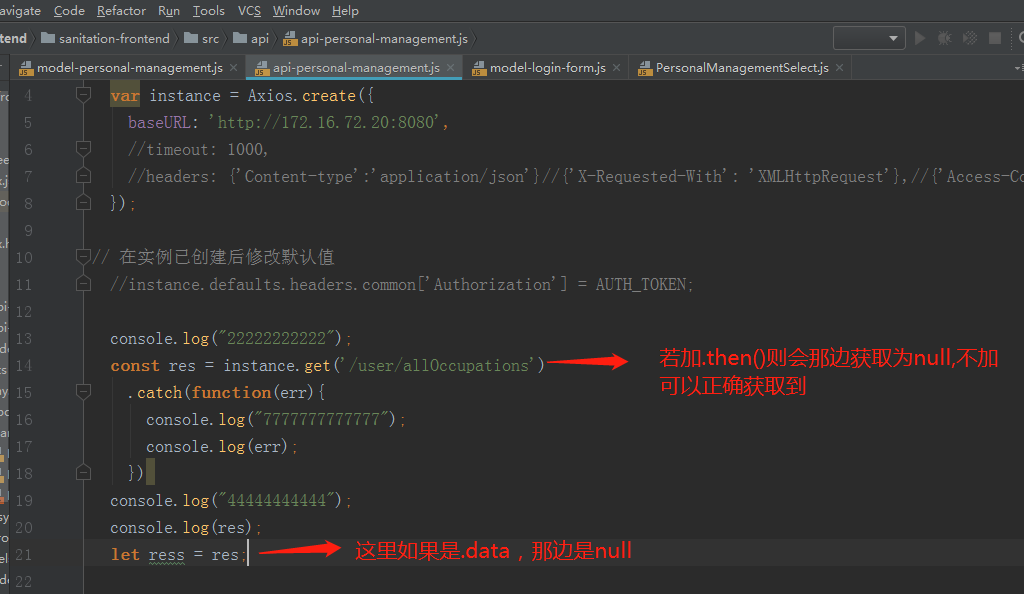
注意：超时配置的默认值是 `0`





/post参数，使用axios.post(url,{},config);如果没有额外的也必须要用一个空对象，否则会报错

1．关于.get函数后，call的返回值有时间延时的问题



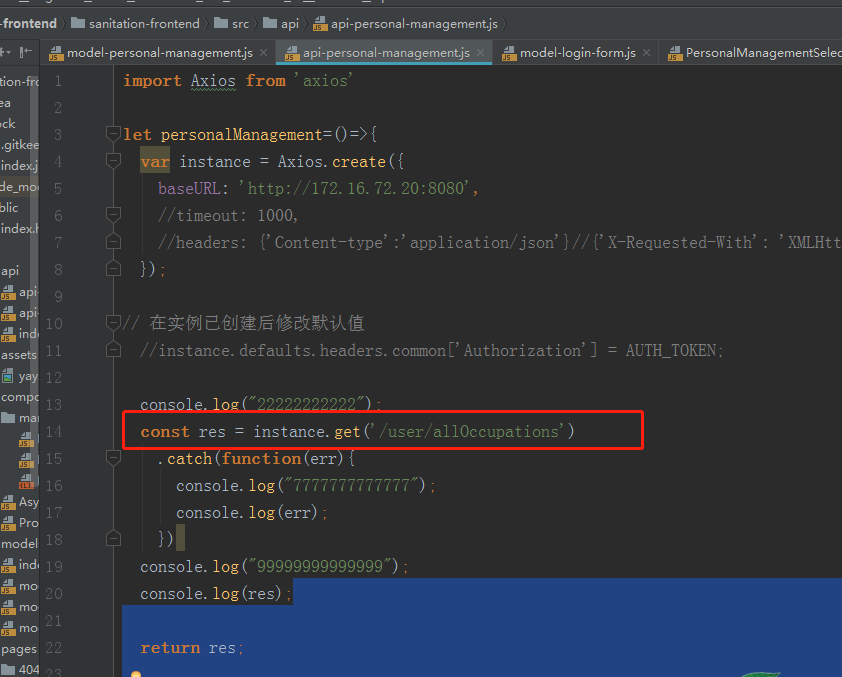
Promise异步数据显示.then这块不懂

<https://segmentfault.com/q/1010000014261353/a-1020000014261614>

<https://www.kancloud.cn/yunye/axios/234845>

<https://segmentfault.com/q/1010000010231720/a-1020000010234539>

1.没有写拦截器，没有设置baseUrl的话，url的地址就要加上http的前缀和端口号:



2.

智能环卫项目的问题：

1. axios的.get函数后获取到的是promise类型的数据，只能在call函数返回后才能取到具体的值，在.get函数那边取具体值会报错
2. 不懂.get后面跟了.then后为啥就异步了，call那边就只能下一次才能获取到数据

Axios缓存

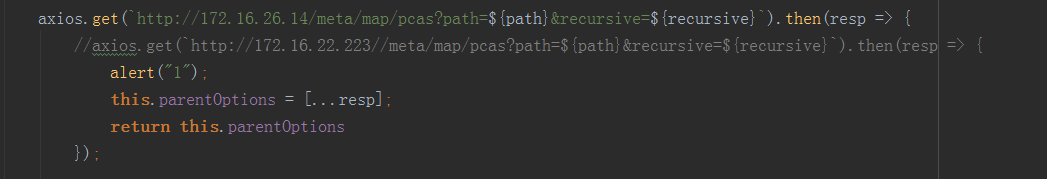
1. <https://segmentfault.com/a/1190000013167994>

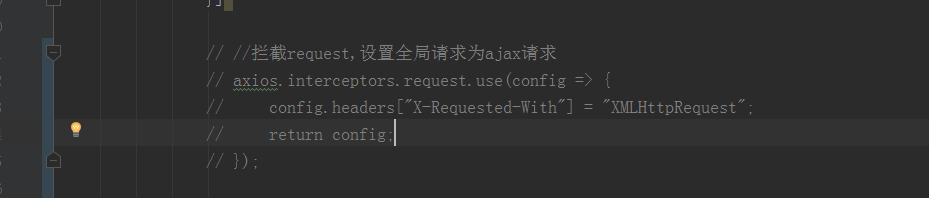


问：为什么不建议永久储存在本地

答：储存在内存中，能保证每次刷新浏览器，加载的都是最新数据；储存在本地，拿不到最新数据；如果需要做到储存在本地，可把每一次数据设置一个版本号，进入应用时向后台发送一个请求，来判断当前请求得到的版本号是否与储存的版本号一致

1. axios的使用：
2. npm install axios –save-dev
3. 引入 import axios from "axios";即可
4. 调用



1. 注意：不要乱加header,否则会改变请求的方式：  
   

注意：这个是跨域问题，要在后台加允许请求：  
