

# 后台数据请求request

封装 request

使用方法

局部封装 (不推荐)

全局封装 (推荐)

## 封装 request

普通的 axios 请求返回的结果如下：

res 的 data 中还要再访问一次 data (res.data.data) 才能得到真正的数据，太过于复杂：

```
1  mounted() { // 页面加载完成之后触发
2    axios.get('http://localhost:9090/user/selectAll').then(res => {
3      console.log(res.data) // 后台返回的数据
4      // res.data = { // 数据格式
5        //   code: '200',
6        //   msg: '请求成功',
7        //   data: {
8          //
9        //   }
10     // }
11   })
12 },
```

而且每次访问一个路由时我们需要后台权限，显然每一次都输入 token 过于麻烦了。

解决方案：在 vue 项目的 src 下创建 utils 目录，随后再创建一个 request.js，将如下代码放入其中：

```
1  import axios from 'axios'
2
3  // 创建可一个新的axios对象
4  const request = axios.create({
5    baseURL: 'http://localhost:9090', // 后端的接口地址 ip:port
6    timeout: 30000
7  })
8
9  // request 拦截器
10 // 可以自请求发送前对请求做一些处理
11 // 比如统一加token, 对请求参数统一加密
12 request.interceptors.request.use(config => {
13   config.headers['Content-Type'] = 'application/json;charset=utf-8';
14   // let user = localStorage.getItem("user") ? JSON.parse(localStorage.getItem("u
15   ser")) : null
16   // config.headers['token'] = 'token' // 设置请求头,通过这种方式发送token
17   return config
18 }, error => {
19   console.error('request error: ' + error) // for debug
20   return Promise.reject(error)
21 });
22
23 // response 拦截器
24 // 可以在接口响应后统一处理结果
25 request.interceptors.response.use(
26   response => {
27     let res = response.data;
28
29     // 兼容服务端返回的字符串数据
30     if (typeof res === 'string') {
31       res = res ? JSON.parse(res) : res
32     }
33     return res;
34   },
35   error => {
36     console.error('response error: ' + error) // for debug
37     return Promise.reject(error)
38   }
39 )
40
41
42 export default request
```

## 使用方法

### 局部封装（不推荐）

在需要访问数据的 vue 文件中添加如下内容：

▼ Vue |

```
1 import request from "@/utils/request";
```

使用语法如下：

▼ Vue |

```
1 mounted() { // 页面加载完成之后触发
2   request.get('/user/selectAll').then(res => {
3     this.users = res.data
4   })
5 },
```

## 全局封装（推荐）

全局封装：在 main.js 中引入\$request 作为全局变量

▼ Vue |

```
1 import request from "@/utils/request";
2
3 Vue.prototype.$request=request
```

使用语法如下：

▼ Vue |

```
1 method() {
2   login(){
3     this.$request.post('/login').then(res => {
4       console.log(res)
5     })
6   }
7
8 },
```