手拉手系列2：在VUE CLI3项目中使用axios

**尤大大在VUE2.0开始就宣告不再继续维护****vue-resource，并推荐大家使用 axios 开始， axios 被越来越多的人所了解。我也是在公司的几个项目中开始用这玩意儿的，之前在公司项目用过vue-resource，总的来说，用起来还可以，既然是尤大大推荐的，那我们还是给点面子用一下，毕竟大神们都在用。  
官方文档**[**https://github.com/axios/axios**](https://github.com/axios/axios)

简单介绍下如何使用axios.

一： 安装

简单：

1. 如果 CDN, 引入<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
2. 如果NPM

* **运行npm install axios**
* **封装axios ， 在untils下面新建一个JS文件http.js**

'use strict'

import axios from 'axios'

import qs from 'qs'

axios.interceptors.request.use(config => { // 这里的config包含每次请求的内容

// 判断localStorage中是否存在api\_token

if (localStorage.getItem('api\_token')) {

// 存在将api\_token写入 request header

config.headers.apiToken = `${localStorage.getItem('api\_token')}`;

}

return config;

}, err => {

return Promise.reject(err);

});

axios.interceptors.response.use(response => {

return response

}, error => {

return Promise.resolve(error.response)

});

function checkStatus (response) {

// 如果http状态码正常，则直接返回数据

if (response && (response.status === 200 || response.status === 304 ||

response.status === 400)) {

return response

}

// 异常状态下，把错误信息返回去

return {

status: -404,

msg: '网络异常'

}

}

function checkCode (res) {

// 如果code异常(这里已经包括网络错误，服务器错误，后端抛出的错误)，可以弹出一个错误提示，告诉用户

if (res.status === -404) {

alert(res.msg)

}

if (res.data && (!res.data.success)) {

// alert(res.data.error\_msg)

}

return res

}

// 请求方式的配置

export default {

post (url, data) { // post

return axios({

method: 'post',

baseURL: '/backapis',

url,

data: qs.stringify(data),

timeout: 5000,

headers: {

'X-Requested-With': 'XMLHttpRequest',

'Content-Type': 'application/x-www-form-urlencoded; charset=UTF-8'

}

}).then(

(response) => {

return checkStatus(response)

}

).then(

(res) => {

return checkCode(res)

}

)

},

get (url, params) { // get

return axios({

method: 'get',

baseURL: '/backapis',

url,

params, // get 请求时带的参数

timeout: 5000,

headers: {

'X-Requested-With': 'XMLHttpRequest'

}

}).then(

(response) => {

return checkStatus(response)

}

).then(

(res) => {

return checkCode(res)

}

)

}

}

* **在main.js中全局引入axios :**

import axios from './untils/http';

Vue.prototype.$axios = axios;

* **现在可以愉快的使用axios了**

loadData(){

const \_this = this

\_this.$axios.get('######').then((rs) => {

\_this.username = rs.data.user

}).catch((err) => {

console.log(err)

});

}