二次封装axios并使用

候鸟与暖风

0.493

2019.08.02 12:03:11

字数 81阅读 5,278

1.新增axios.js文件，内容如下（二次封装axios,方便我们后续项目的使用）

参数的数据格式分两种

1.json对象类型  {name:1,password:1356565}

2.字符串类型    name:1,password:1356598   (需要下载qs插件进行转换)

import axios from 'axios'

import Qs from 'qs'（如果只是get请求就不需要引qs）

import { Message } from 'element-ui'  //引入element中的弹窗，这个不是必须的可根据项目来

axios.defaults.withCredentials = false;  //false  不跨域  true 跨域

axios.defaults.headers.common['token'] = localstorage.getItem('token'); //设置请求头，这个是不是必须的

export  default *function*(*path*, *params*, *method* = "GET", *headerType* = "json")=>{

*let* baseURL = "", data = {}

    baseURL = process.env.BASE\_URL //这里是在config里面配置了请求路径

      //设置请求头

    if (method === 'post') {   //post请求

        if (headerType === "json") {

            axios.defaults.headers["Content-Type"] = 'application/json;charset=UTF-8'

            data = params

        } else {

            axios.defaults.headers["Content-Type"] = 'application/x-www-form-urlencoded;charset=UTF-8'

            data = Qs.stringify(params);

        }

    }

    //get请求

    if (method === 'get') {

        if (headerType == !'json') {

            axios.defaults.headers["Content-Type"] = 'application/x-www-form-urlencoded;charset=UTF-8'

        }

        data = Qs.stringify(params);

        path = path + '?' + data

        data = {};

    }

     // 请求拦截

    axios.interceptors.request.use(*function* (*config*) {

*let* token = storage.getStorage("token")

        if (token) {

*config*.headers['token'] = token;

        }

        return *config*;

       }, *function* (*error*) {

        return *Promise*.reject(*error*);

       });

      //响应拦截

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

          if (*response*.headers.token) {

            storage.setStorage('token', *response*.headers.token)

          }

          if (*response*.data.statusCode == 20009) {  //这里的状态码是根据后台设置的来

               Message.error({ message: '登录过期，请重新登录' })

           }

           return *response*;

           },*error=>*{

              return *Promise*.resolve(*error*.response)

        })

       //发送请求

      return new *Promise*(()*=>*{

         axios({

            baseURL

            method

            url: path,

            headers，

            data:params,

            timeout:6000,

            }).then(*result=>*{

                resolve(*result*.data)

            }).catch(*err=>*{

                reject(*err*)

            })

         })

}

2.新增api.js文件，内容如下（项目中所有的请求地址以及方式等都放在这个文件中，并导出）

import http from '@/utils/axios'  //引入我们二次封装的axios.js文件

a. post请求，参数是json类型

          export *const* passwordUpdate = *function* (*params*) {

*const* url= '/account/accountUpdate';  //请求的地址

                return https(PASSWORDUPDARE, *params*, "post");

          };

b.post请求，参数是字符串类型

           export *const* passwordUpdate = *function* (*params*) {

*const* url= "/account/accountUpdate";  //请求的地址

                return https(PASSWORDUPDARE , *params*, "post", "from");

            };

c.get请求

           export *const* passwordUpdate = *function* (*params*) {

*const* url= "/account/accountUpdate";  //请求的地址

                return https(PASSWORDUPDARE , *params*);

            };

3.页面中使用我们封装好的请求

1.先引入：import { requestLogin } from '@/api/api'  //需要调用哪个接口，就引入哪个，按需引入

2.使用

*let* params = {

  name:'zhangsan',

  password:'123456'

}

requestLogin(params ).then(*res=>*{   //我们只需要传递一个参数，其他的都已经在api.js中写好了

    //获取到后台响应的参数res，并进行操作

})