



# config形式

axios({

method:’get’,// 默认get方式；

url:’url’

).then( (response) =>{

console.log(response.data)

).catch( (error)=>{

console.log(error)

});

# 方法调用

axios.get({’url’,{params:{

age:20

}

}).then( (response) =>{

console.log(response.data)

).catch( (error)=>{

console.log(error)

});

axios.post({’url’,{

age:20

}).then( (response) =>{

console.log(response.data)

).catch( (error)=>{

console.log(error)

});

# 全局配置信息

axios对象的默认配置



import axios from ‘axios’;

import queryString from ‘queryString’;

//创建取消请求的令牌

var CancelToken = axios.CancelTonken;

var source = CancelToken.source();

var HTTP= axios.create({

baseURL:’’,

timeout: 1000, //毫秒

responseType: ‘json’, //response数据类型

params:{//无论请求格式都会以查询字符串的形式出现在请求地址上

age:20

},

headers:{//请求头信息

‘custome-header’: 20,

‘content-type’:’application/x=www=form-urlencoded’//做了transformRequest的数据转换为字符串后，如果是post请求需要添加；

},

transformRequest:[function(data){//转换请求数据内容或者格式,支持多个函数多次转换；

//这里的data默认是一个json对象，使用queryString转换成字符串；

//转换的格式为’name=val&name=val’

console.log(data)

return queryString.stringify(data); //一定要返回数据（返回数据会转换成字符串），否则在network的请求数据没有显示；因为函数默认返回值是undefined；但是请求数据能正常返回；

}],

transformResponse:[function(data){

console.log(data);

return queryString.stringify(data); //一定要返回数据,否则请求返回数据为undefined；因为函数默认返回值是undefined；

}],

validateStatus(status){//请求是否成功；影响premise执行then或者catch

console.log(status);

return true; //返回true为请求成功；

},

cancelToken:source.token //需要上面取消请求的令牌配合，具体返回和调用见，取消请求

})

## 取消请求

需要config的cancelToken配置

调用取消，会执行promise的catch；

axios.post({’url’,{

age:20

}).then( (response) =>{

console.log(response.data)

).catch( (error)=>{

if( axios.isCancel(error)){//区分是否是用户取消的请求

console.log(error.message)

}else{

console.log(error)

}

});

source.concel(‘用户取消请求’) //调用即可取消；

## 全局请求拦截

HTTP.interceptors.request.use(function (config) {//config之前的配置信息，如果没有就是默认

//在发送请求之前做些事

return config //必须返回请求的配置信息，否则拦截请求

}),function(){

//请求错误时做些事

return Promise.reject(error)

}

## 全局响应

HTTP.interceptors.response.use(function (data) {//data请求返回的数据

//请求成功做些事

return data//必须返回data，否则请求返回数据为undefined；因为函数默认返回值是undefined；

})

# 并发请求

axios.spread(callback)

axios.all(iterable)

同时送多个请求，都成功才会请求成功；返回数组对应请求个数可以使用axios.spread(callback)进行展开； 用法如下:

function http1(){

return axios.get(‘url’)

}

function http2(){

return axios.post(‘url’)

}

axios.all([http1(),http2()])

.then( axios.spread(res1, res2){

console.log(res1);

console.log(res2);

}).catch( (error)=>{

if( axios.isCancel(error)){//区分是否是用户取消的请求

console.log(error.message)

}else{

console.log(error)

}

});

# 在vue中使用

安装

npm install axios vue-axios --save

作为插件

import VueAxios from ‘ vue-axios’

import Axios from ‘ axios’

Vue.use(VueAxios,Axios)

在组件中使用

this.$http[method]();