

在vue中选择什么网络请求模块？

Vue中发送网络请求有非常多的方式, 那么, 在开发中, 如何选择呢?

选择一: 传统的Ajax是基于XMLHttpRequest(XHR)

为什么不用它呢?

非常好解释, 配置和调用方式等非常混乱.

编码起来看起来就非常繁琐.

所以真实开发中很少直接使用, 而是使用jQuery-Ajax

选择二: 在使用Vue之前, 我们经常会使用jQuery-Ajax

相对于传统的Ajax非常好用.

为什么不选择它呢?

首先, 我们先明确一点: 在Vue的整个开发中都是不需要使用jQuery了.

那么, 就意味着为了方便我们进行一个网络请求, 特意引用一个jQuery, 你觉得合理吗?

jQuery的代码1w+行.

Vue的代码才1w+行.

完全没有必要为了用网络请求就引用这个重量级的框架

选择三: 官方在Vue1.x的时候, 推出了Vue-resource.

Vue-resource的体积相对于jQuery小很多.

另外Vue-resource是官方推出的.

为什么不选择它呢?

在Vue2.0退出后, Vue作者就在GitHub的Issues中说明了去掉vue-resource, 并且以后也不会再更新.

那么意味着以后vue-resource不再支持新的版本时, 也不会再继续更新和维护.

对以后的项目开发和维护都存在很大的隐患.

选择四: 在说明不再继续更新和维护vue-resource的同时, 作者还推荐了一个框架: axios为什么不用它呢?

axios有非常多的优点, 并且用起来也非常方便.

稍后, 我们对他进行详细学习.

为什么选择axios?

功能特点:

- 1.在浏览器中发送 XMLHttpRequests 请求
 - 2.在 node.js 中发送 http请求
 - 3.支持 Promise API
 - 4.拦截请求和响应
 - 5.转换请求和响应数据
- 等等

axios

测试接口

| 接口名 | 请求方式 | 请求参数 | 请求地址 |
|---------|------|-------------------------|----------------------|
| 历史上的今天 | get | 无参 | href |
| 垃圾分类查询 | get | m string | href |
| 天气查询 | post | city string type string | href |
| 获取公历与农历 | post | 无参 | href |

axios

axios是基于promise对ajax的一种封装

ajax mvc

axios mvvm

在使用axios之前，需要在Vue-CLI中先去安装它

npm install axios --save

在要使用的组件中直接引入调用即可

import axios from 'axios'

axios的基本使用

axios默认请求

```
//使用默认方式发送无参请求
import axios from "axios";
axios({
  url: 'https://api.muxiaoguo.cn/api/lishijr'
}).then(res => {
  console.log(res)
})
//默认使用get方式进行请求
```

axios的响应结构

//与jquery中的ajax不同的是，axios会对返回结果进行封装，将其封装为如下结构：

```
{
  data:{},           //后台返回的数据
  status:200,        //状态码
  statusText: 'OK',  //状态信息
  headers:{},        //服务器头部信息
  config:{},         //axios对request的配置
  request:{}          //request对象
}
```

axios的请求方式

axios发送get请求

//指定方式为get的无参请求

```
import axios from "axios";
axios({
  url: 'https://api.muxiaoguo.cn/api/lishijr',
  method: 'get'
}).then(res => {
  console.log(res)
})
```

//使用get方式发送带有参数的请求

```
import axios from "axios";
axios({
  url: 'https://api.muxiaoguo.cn/api/lajifl?m=啤酒',
  method: 'get'
}).then(res => {
  console.log(res)
})
```

//使用get方式发送有参请求其他形式

```
import axios from "axios";
axios({
  url: 'https://api.muxiaoguo.cn/api/lajifl',
  params: {
    m: '啤酒'
  },
  method: 'get'
}).then(res => {
  console.log(res)
})
```

axios发送post请求

//使用post方式发送无参请求

```
import axios from "axios";
axios({
  url: 'https://api.muxiaoguo.cn/api/yinlongli',
  method: 'post'
}).then(res => {
  console.log(res)
})
```

//使用post方式发送带有参数的请求 - 通过params

```
import axios from "axios";
axios({
  url: 'https://api.muxiaoguo.cn/api/tianqi',
  params: {
    city: '柳州',
    type: 1 //url上拼接参数, 明文拼接, 不推荐使用
  },
  method: 'post'
}).then(res => {
  console.log(res)
})
```

//使用post方式发送带有参数的请求 使用data传递

```
import axios from "axios";
axios({
  url: 'https://api.muxiaoguo.cn/api/tianqi',
  data: {
    ciry: '柳州',
    type: 1 //后台接收为null
  },
  method: 'post'
}).then(res => {
  console.log(res)
})
```

```
})
```

//当使用data传递的时候，后台接收到的参数为null；axios使用post携带参数请求默认使用application/json格式

//解决方式1：使用params属性进行数据的传递 不推荐 明文拼接

//解决方式2：axios.post 后携带参数 "page=1&pageSize=10" 与 url 进行拼接

//解决方式3：服务器端给接收的参数加上@requestBody, 或进行其他配置

//解决方式4：配置axios的headers，使用qs序列化参数，将json或者数组对象序列化成url的形式，以&进行拼接

```
<!-- 安装qs序列化工具 npm install qs --save -->
```

// post 有参

```
import axios from "axios";
```

```
import qs from "qs";
```

```
axios({
```

```
  url: 'https://api.muxiaoguo.cn/api/tianqi',
```

```
  method: 'post',
```

```
  data: qs.stringify({
```

```
    city: '柳州',
```

```
    type: 1
```

```
  })),
```

```
  headers: {
```

```
    'Content-Type': 'application/x-www-form-urlencoded;
```

```
charset=UTF-8'
```

```
  }
```

```
}).then(r => {
```

```
  console.log('post有参', r);
```

```
})
```

axios快捷请求方式

//使用axios.get方式发送无参请求 .catch报错处理

```
import axios from "axios";
axios.get('https://api.muxiaoguo.cn/api/lishijr').then(res
=> {
  console.log(res)
}).catch(error => {
  console.log('报错啦')
  console.log(error)
})
```

//使用axios.get发送有参请求 - 拼接参数

```
import axios from "axios";
axios.get('https://api.muxiaoguo.cn/api/lajifl?m=啤
酒').then(res => {
  console.log(res);
})
```

//使用axios.get发送有参请求 - params

```
import axios from "axios";
axios.get('https://api.muxiaoguo.cn/api/lajifl',
{params:{m:'啤酒'}}).then(res => {
  console.log(res)
}).catch(error => {
  console.log('报错啦')
  console.log(error)
})
```


//使用axios.post发送无参请求

```
import axios from "axios";
axios.post('https://api.muxiaoguo.cn/api/yinlongli').then(
res => {
  console.log(res)
}).catch(error => {
  console.log('报错啦')
  console.log(error)
})
```

//使用axios.post发送有参请求

//发送post请求携带参数，直接使用"page=1&pageSize=10"

```
import axios from "axios";
axios.post('https://api.muxiaoguo.cn/api/tianqi','city='柳州'&type=1')
.then(res => {
  console.log(res)
}).catch(error => {
  console.log('报错啦')
  console.log(error)
})
```

//使用data传递数据，后台需要将axios自动转换的json数据转换为java对象，修改后台代码

//接收的参数添加@requestBody注解

```
import axios from "axios";
axios.post('https://api.muxiaoguo.cn/api/tianqi',{city:'柳州',type:1})
.then(res => {
  console.log(res)
}).catch(error => {
  console.log('报错啦')
  console.log(error)
})
```

```
})
```

//配置axios的headers, 使用qs序列化参数, 将json或者数组对象序列化成url的形式, 以&进行拼接

// post有参

```
import axios from "axios";
```

```
import qs from "qs";
```

```
axios.post('https://api.muxiaoguo.cn/api/tianqi',
```

```
qs.stringify({
```

```
  city: '柳州',
```

```
  type: 1
```

```
}),
```

```
{
```

```
  headers: {
```

```
    'Content-Type': 'application/x-www-form-urlencoded;
```

```
charset=UTF-8'
```

```
  }
```

```
}
```

```
)
```

```
.then(r => {
```

```
  console.log('post有参', r);
```

```
})
```

axios并发请求

//axios的并发请求

```
import axios from "axios";
```

// axios并发请求

```
axios.all([
```

```
  axios.get('https://api.muxiaoguo.cn/api/lishijr'),
```

```
  axios.get('https://api.muxiaoguo.cn/api/lajifl',
```

```
    {params:{m: '啤酒'}}})
]).then(res => { //请求成功响应的是一个数组
    // 注意 只要一个接口有问题 所有的数据都拿不到
    console.log(res[0])
    console.log(res[1])
}).catch(err => {
    console.log(err)
})
```

```
//使用spread方法处理响应结果
// axios并发请求
import axios from "axios";
axios.all([
    axios.get('https://api.muxiaoguo.cn/api/lishijr'),
    axios.get('https://api.muxiaoguo.cn/api/lajifl',
        {params:{m: '啤酒'}})
]).then(
    axios.spread((res1, res2) => {
        console.log(res1)
        console.log(res2)
    })
).catch(err => {
    console.log(err)
})
```

axios的全局配置

```
import axios from "axios";
// 全局配置
// 设置全局属性baseUrl
```

```
axios.defaults.baseURL =  
'http://47.106.244.1:8099/manager';  
// 设置超时时间 不要设置的太低 比如设置5 5ms 一般都会报错  
axios.defaults.timeout = 5000;  
  
//在全局配置的基础上请求  
axios.get('category/findAllCategory').then(res => {  
  console.log(res)  
}).catch(err => {  
  console.log(err)  
})  
  
axios.post('user/logout').then(res => {  
  console.log(res)  
}).catch(err => {  
  console.log(err)  
})
```

axios的拦截器

axios给我们提供了两大类拦截器

一种是请求方向的拦截（成功请求，失败请求）

另一种是响应方向的（成功响应，失败响应）

拦截器的作用：用于我们在网络请求的时候在发起请求或者响应时对操作进行相应的处理

例如：发起请求时可以添加网页加载的动画 强制登录

响应的时候可以进行相应的数据处理

//请求方向拦截器

```
import axios from "axios";
// 拦截器
axios.interceptors.request.use(config => {
  // 可以在请求拦截器里做什么呢?
  // 1.当用户发送网络请求的时候 在页面中添加一个loading组件 作为动画
  // 2.用户登录的时候 判断用户是否有token 有的话就跳转到界面 没有的话就返回到登录界面或提示报错
  // 等等。。。
  console.log('进入请求拦截器')
  console.log(config)
  return config //记得return, 放行请求
}, err => {
  console.log('请求方向失败')
  console.log(err)
})

axios.get('https://api.muxiaoguo.cn/api/lishijr').then(res
=> {
  console.log(res)
})
```

//响应方向拦截器

```
import axios from "axios";
axios.interceptors.response.use(config => {
  //可以在响应拦截器里做什么呢?
  //可以拿到请求响应的状态码 根据状态码去做一些判断
  console.log('状态码', config.data.status);
  console.log('进入响应拦截器')
  return config.data //只会返回data
}, err => {
  console.log('响应方向失败')
  console.log(err)
})
```

```
})
```

```
axios.get('https://api.muxiaoguo.cn/api/lishijr').then(res  
=> {  
  console.log(res)  
})
```