

网络请求库 – axios库

王红元 coderwhy

目录

content



1 认识Axios库

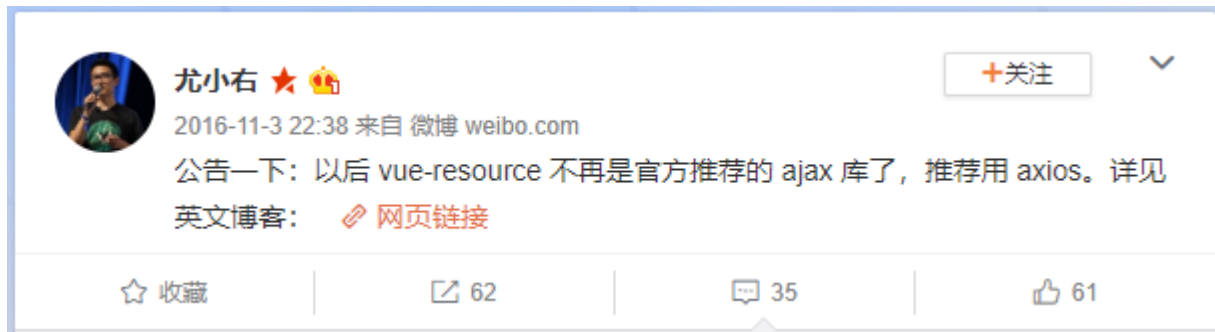
2 axios发送请求

3 axios创建实例

4 axios的拦截器

5 axios请求封装

■ 为什么选择axios? 作者推荐和功能特点



■ 功能特点:

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

■ 补充: axios名称的由来? 个人理解

- 没有具体的翻译.
- axios: ajax i/o system.

■ 支持多种请求方式:

- `axios(config)`
- `axios.request(config)`
- `axios.get(url[, config])`
- `axios.delete(url[, config])`
- `axios.head(url[, config])`
- `axios.post(url[, data[, config]])`
- `axios.put(url[, data[, config]])`
- `axios.patch(url[, data[, config]])`

■ 有时候, 我们可能需求同时发送两个请求

- 使用`axios.all`, 可以放入多个请求的数组.
- `axios.all([])` 返回的结果是一个数组, 使用 `axios.spread` 可将数组 `[res1,res2]` 展开为 `res1, res2`

常见的配置选项

■ 请求地址

- url: '/user',

■ 请求类型

- method: 'get',

■ 请根路径

- baseUrl: 'http://www.mt.com/api',

■ 请求前的数据处理

- transformRequest:[function(data){}],

■ 请求后的数据处理

- transformResponse: [function(data){}],

■ 自定义的请求头

- headers:{'x-Requested-With':'XMLHttpRequest'},

■ URL查询对象

- params:{ id: 12 },

■ 查询对象序列化函数

- paramsSerializer: function(params){ }

■ request body

- data: { key: 'aa'},

■ 超时设置

- timeout: 1000,

axios的创建实例

■ 为什么要创建axios的实例呢?

- 当我们从axios模块中导入对象时, 使用的实例是默认的实例;
- 当给该实例设置一些默认配置时, 这些配置就被固定下来了.
- 但是后续开发中, 某些配置可能会不太一样;
- 比如某些请求需要使用特定的baseURL或者timeout等.
- 这个时候, 我们就可以创建新的实例, 并且传入属于该实例的配置信息.

```
const instance = axios.create({
  baseURL: "http://123.207.32.32:1888"
})

instance.post("/02_param/postjson", {
  name: "why",
  age: 18
}).then(res => {
  console.log("res:", res)
})
```

请求和响应拦截器

■ axios的也可以设置拦截器：拦截每次请求和响应

- ❑ axios.interceptors.request.use(请求成功拦截, 请求失败拦截)
- ❑ axios.interceptors.response.use(响应成功拦截, 响应失败拦截)

```
axios.interceptors.request.use((config) => {  
  console.log("请求成功拦截")  
  return config  
}, err => {  
  console.log("请求失败拦截")  
  return err  
})  
  
axios.interceptors.response.use((res) => {  
  console.log("响应成功拦截")  
  return res.data  
}, err => {  
  console.log("响应失败失败")  
})
```

axios请求库封装（简洁版）

```
class HYRequest {  
  constructor(baseUrl) {  
    this.instance = axios.create({ ...  
  })  
}  
  
  request(config) {  
    return new Promise((resolve, reject) => {  
      this.instance.request(config).then(res => {  
        resolve(res.data)  
      }).catch(err => {  
        console.log("request err:", err)  
        reject(err)  
      })  
    })  
  }  
  
  get(config) {  
    return this.request({ ...config, method: "get" })  
  }  
  
  post(config) {  
    return this.request({ ...config, method: "post" })  
  }  
}
```