

1. 文档

<https://github.com/axios/axios>

2. 是什么

前端最流行的ajax请求库
react/vue官方都推荐使用axios发ajax请求

3. axios的特点

基本promise的异步ajax请求库
浏览器端/node端都可以使用
支持请求 / 响应拦截器
支持请求取消
请求/响应数据转换
批量发送多个请求

4. axios常用语法

axios(config): 通用/最本质的发任意类型请求的方式
axios(url[, config]): 可以只指定url发get请求
axios.request(config): 等同于axios(config)
axios.get(url[, config]): 发get请求
axios.delete(url[, config]): 发delete请求
axios.post(url[, data, config]): 发post请求
axios.put(url[, data, config]): 发put请求

axios.defaults.xxx: 请求的默认全局配置
axios.interceptors.request.use(): 添加请求拦截器
axios.interceptors.response.use(): 添加响应拦截器

axios.create([config]): 创建一个新的axios(它没有下面的功能)

axios.Cancel(): 用于创建取消请求的错误对象
axios.CancelToken(): 用于创建取消请求的token对象
axios.isCancel(): 是否是一个取消请求的错误
axios.all(promises): 用于批量执行多个异步请求
axios.spread(): 用来指定接收所有成功数据的回调函数的方法

5. 难点语法理解与使用

- 1). axios.create(config)
 - a. 根据指定配置创建一个新的axios, 也就每个新axios都有自己的配置

- b. 新axios只是没有取消请求和批量发请求的方法，其它所有语法都是一致的
- c. 为什么要设计这个语法?
 - 需求：项目中有部分接口需要的配置与另一部分接口需要的配置不太一样，如何处理
 - 解决：创建2个新axios，每个都有自己特有的配置，分别应用到不同要求的接口请求中

2). 拦截器函数/ajax请求/请求的回调函数的调用顺序

- a. 说明：调用axios()并不是立即发送ajax请求，而是需要经历一个较长的流程
- b. 流程：请求拦截器2 => 请求拦截器1 => 发ajax请求 => 响应拦截器1 => 响应拦截器2 => 请求的回调
- c. 注意：此流程是通过promise串连起来的，请求拦截器传递的是config，响应拦截器传递的是response
错误流程控制与错误处理

3). 取消请求

- a. 基本流程：
 - 配置cancelToken对象
 - 缓存用于取消请求的cancel函数
 - 在后面特定时机调用cancel函数取消请求
 - 在错误回调中判断如果error是cancel，做相应处理
- b. 实现功能
 - 点击按钮，取消某个正在请求中的请求
 - 在请求一个接口前，取消前面一个未完成的请求