结构

FAQ:

Promise.all中出错的请求对其他请求的影响

Promise.all的执行顺序和返回顺序

李白和星星

于 2022-02-25 12:22:14 发布

3021

收藏 3

文章标签： javascript vue.js 前端 es6

版权

Promise.all()用法：多个接口全部返回值后再进行下一步操作

Promise.all([a(),b()]).then(res=>{

console.log(res)})

最近项目中有个问题 需要同时返回a和b接口的返回值再做下一步 并且要对不同的返回值做不同的处理 这就需要区分res的返回值中究竟是否有返回顺序

zsgroupMemberApply(qs.stringify({ checkResult: 0 }) ).then(res=>{

console.log('A')

})

zjgroupMemberApply(qs.stringify({ checkResult: 0 }) ).then(res=>{

console.log('B')

})

Promise.all([

zsgroupMemberApply(qs.stringify({ checkResult: 0 })),

zjgroupMemberApply(qs.stringify({ checkResult: 0 })),

]).then((res) => {

console.log(res)

var list1=res[0].data.data.list;

var list2=res[1].data.data.list;

for(var i=0;i<list1.length;i++){

list1[i].f\_type='zs'

}

for(var j=0;j<list2.length;j++){

list2[j].f\_type='zj'

}

this.list =list1.concat(list2);

});

打印结果

B

A

[A,B]

**由此可见 Promise.all()里面接口的执行顺序是按接口的并行执行来的 但是返回值依然是按照其接口的代码的先后顺序来的**

————————————————

版权声明：本文为CSDN博主「李白和星星」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/qq_35469137/article/details/123130179>

# **await 与 Promise.all 结合使用**

[](https://www.jianshu.com/u/9a55f12c13d9)

[yuho\_小豪哥](https://www.jianshu.com/u/9a55f12c13d9" \t "https://www.jianshu.com/p/_blank)关注

0.322019.11.08 10:03:12字数 169阅读 18,305

当遇到多个可以同时执行的异步任务时，就需要使用 Promise.all。

Promise.all 方法用于将多个 Promise 实例，包装成一个新的 Promise 实例。

const p = Promise.all([p1, p2, p3])

Promise.all 方法接受一个数组作为参数，p1、p2、p3 都是 Promise 实例，如果不是，就会先调用 Promise.resolve 方法，将参数转为 Promise 实例，再进一步处理。（Promise.all 方法的参数可以不是数组，但必须具有 Iterator 接口，且返回的每个成员都是 Promise 实例。）

而 async/await 本身就是 promise 的语法糖，因此可以与 Promise.all 结合使用：

const p1 = async () => {}const p2 = async () => {}const p3 = async () => {}

const [result1, result2, result3] = await Promise.all([p1, p2, p3])

console.log(result1)

console.log(result2)

console.log(result3)