

Promise的错误处理

发布于2020-06-16 15:31:09 阅读 874



题图 By CIm

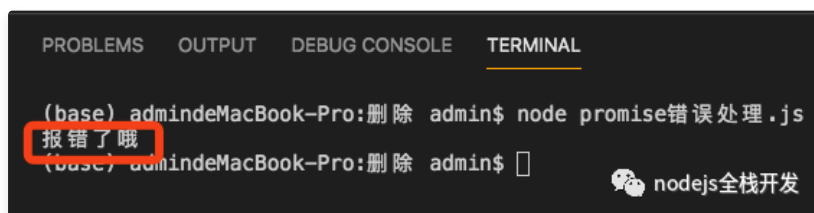
在开发过程中我们经常使用Promise来处理异步，但是我们经常忽略Promise的错误处理。

今天带着大家一起来梳理一下Promise处理错误的几种情况。

第一种情况是直接抛出error，在Promise中抛出错误只有throw和reject这两种方式，并且throw和reject抛出错误在Promise中没有区别，这两种方式都可以被catch所捕获，先看throw，代码如下：

```
1 | new Promise((resolve, reject) => {  
2 |   throw "报错了哦"  
3 | }).catch((e) => {  
4 |   console.log(e)  
5 | })
```

打印如图：



再看通过reject来抛出错误，代码如下：

```
1 | new Promise((resolve, reject) => {  
2 |   reject("报错了哦")  
3 | }).catch((e) => {  
4 |   console.log(e)
```

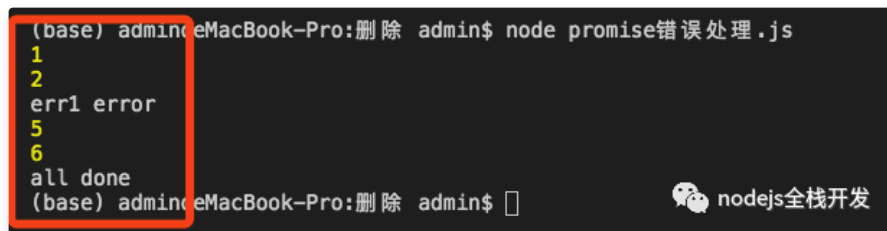
```
5 |   })
6 |
7 |   Promise.reject("报错了哦").catch(e=>{
8 |     console.log(e)
9 |   })
```

阅读源码发现throw和reject抛出的错误都会被catch所捕获，并且没有什么不同。

第二种情况，处理Promise调用链中的错误，如果有一个较长的Promise调用链，其中某个环节抛出错误，错误会被后续链中最近的一个catch所捕获，代码如下：

```
1 | Promise.resolve(1)
2 |   .then(a => console.log(1))
3 |   .then(a => console.log(2))
4 |   .then(a => Promise.reject('error'))
5 |   .then(a => console.log(3))
6 |   .then(a => console.log(4))
7 |   .catch(err => console.log('err1', err))
8 |   .then(a => console.log(5))
9 |   .then(a => console.log(6))
10 |  .catch(err => console.log('err2', err))
11 |  .then(() => console.log('all done'))
```

打印结果如下：



```
(base) admin@MacBook-Pro:删除 admin$ node promise错误处理.js
1
2
err1 error
5
6
all done
(base) admin@MacBook-Pro:删除 admin$
```

阅读源码，我们发现调用链中第一个错误，被第一个catch函数所捕获，仔细观察打印结果，发现第一个错误发生的调用链后面的3、4被跳过了。

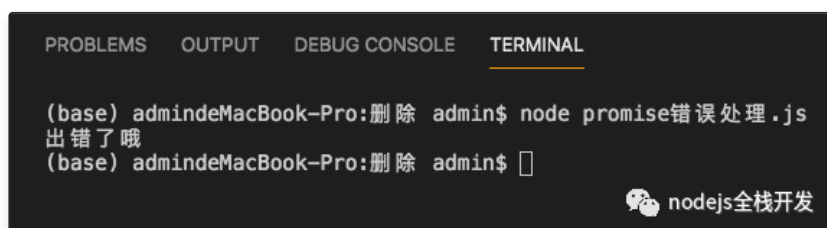
这是Promise调用链的一个特性，调用链中一旦发生错误，这个错误调用链后面的then链会被跳过，直接进入catch函数。

再仔细观察，我们发现第一个catch后面的then依然可以正常执行，这里大家需要注意，Promise调用链中发生错误后，会跳过后面的then链，进入catch函数，但是catch后面的then函数依然可以正常执行。

第三种情况，Promise.all的异常，一般处理并发的时候，我们需要使用Promise.all，但是如果all中有一个Promise实例出现异常，会导致全部结果被丢弃。如果这里的结果不符合你的预期一定要小心处理，看代码：

```
1 | const tasks = [
2 |   Promise.resolve(1),
3 |   Promise.resolve(2),
4 |   Promise.reject("出错了哦"),
5 |   Promise.resolve(4),
6 |   Promise.resolve(5),
7 | ];
8 |
9 | Promise.all(tasks)
10 |   .then(arr => console.log(arr))
11 |   .catch(err => console.log(err))
```

打印结果如下：



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
(base) admin@MacBook-Pro:删除 admin$ node promise错误处理.js
出错了哦
(base) admin@MacBook-Pro:删除 admin$
```

在这个例子中，我们虽然捕获了错误，但是其他结果全部被丢弃了，有的时候这不符合我们的预期，假如我们需要所有的结果，不管错误的还是正确的，该如何做呢？只需要对tasks数组中每一个promise实例都用catch处理一下，代码如下：

```

1 | const tasks = [
2 |   Promise.resolve(1),
3 |   Promise.resolve(2),
4 |   Promise.reject("出错了哦"),
5 |   Promise.resolve(4),
6 |   Promise.resolve(5),
7 | ];
8 |
9 | const taskCatch = tasks.map(e=>e.catch(e=>e))
10 |
11 | Promise.all(taskCatch)
12 |   .then(arr => console.log(arr))
13 |   .catch(err => console.log(err))

```

打印结果如下:

```

^[[A^[[A(base) admin@MacBook-Pro:删除 admin$ node promise错误处理.js
[ 1, 2, '出错了哦', 4, 5 ]
(base) admin@MacBook-Pro:删除 admin$

```

有的同学看到这里可能会想到Promise.race的错误该如何处理，Promise.race虽然也是并发，但是其机制是只取一个结果，这个结果会按照正常的错误机制被捕获，如果做了处理错误处理，其执行路径会发生偏差，这主要是看代码的需求，看代码：

```

1 | const tasks = [
2 |   Promise.reject("出错了哦"),
3 |   Promise.resolve(1),
4 |   Promise.resolve(2),
5 |   Promise.resolve(4),
6 |   Promise.resolve(5),
7 | ];
8 |
9 |
10 | Promise.race(tasks)
11 |   .then(arr => {
12 |     console.log("执行的是then")
13 |     console.log(arr)
14 |   })
15 |   .catch(err => {
16 |     console.log("执行的是catch")
17 |     console.log(err)
18 |   })

```

执行结果如图：

```

(base) admin@MacBook-Pro:删除 admin$ node promise错误处理.js
执行的是 catch
出错了哦
(base) admin@MacBook-Pro:删除 admin$

```

我们看到错误被Promise链最后的catch函数所捕获。我们将代码修改一下，用catch处理tasks中每一个Promise实例。代码如下：


```

1 | const tasks = [
2 |   Promise.reject("出错了哦"),
3 |   Promise.resolve(1),
4 |   Promise.resolve(2),
5 |   Promise.resolve(4),
6 |   Promise.resolve(5),
7 | ];
8 |
9 | const taskCatch = tasks.map(e=>e.catch(e=>e))
10 |
11 | Promise.race(taskCatch)
12 |   .then(arr => {
13 |     console.log("执行的是then")
14 |     console.log(arr)
15 |   })
16 |   .catch(err => {
17 |     console.log("执行的是catch")
18 |   })

```

```
17 |         console.log(err)  
    |     })
```

执行结果：

```
(base) admindeMacBook-Pro:删除 admin$ node promise错误处理.js  
执行的是 then  
出错了哦  
(base) admindeMacBook-Pro:删除 admin$ █  nodejs全栈开发
```

看结果，执行的是then，说明错误在第一个catch函数处被捕获了，catch后面的then链正常执行，所以执行了then函数。

以上便是在使用Promise时需要对错误处理时注意的地方，希望对你有所帮助。

引用资料：

Promise 错误处理

<https://www.52cik.com/2018/04/30/promise-error.html>

举报