

NodeJS 回调的错误处理方式及其优点

作者: [Vanessa](#)

原文链接: <https://ld246.com/article/1552875246782>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

2019-03-18

回答

NodeJS 通常使用回调模式，如果在执行期间发生错误，会把错误作为回调的第一个参数传递到回调函数中：

```
fs.readFile(filePath, function(err, data) {
  if (err) {
    // 错误处理，return 可以让执行停止，这非常重要
    return console.log(err)
  }
  // 使用数据对象
  console.log(data)
})
```

优点包括如下：

- 如果不需要引用数据，则无需对数据进行处理
- API 保持高度的一致性可以带来更多的便捷
- 能够轻松适配回调模式，从而实现更易于维护的代码

从下面的示例中可以看出，如果没有错误，回调函数中的第一个参数将为 null。但是如果出现错误的，将会创建一个 Error 对象作为回调函数中的唯一参数。回调函数让用户更加容易的知道是否有错误生。

这种实践也称为 Node.js 的错误约定，这种回调函数的实现称为错误优先回调。

```
var isTrue = function(value, callback) {
  if (value === true) {
    callback(null, "Value was true.")
  } else {
    callback(new Error("Value is not true!"))
  }
}
```

```
var callback = function(error, data) {
  if (error) {
    console.log(error)
    return
  }
  console.log(data)
}
```

```
isTrue(false, callback) // Error: Value is not true!
isTrue(true, callback) // Value was true.
```

加分回答

- 虽然这只是一种约定，但是我们应该按照约定的去做。

返回总目录

[30 秒面试系列一](#)