

• 谈谈对Promise的理解

■ 前置知识

■ 同步

同步的思想是：所有的操作都做完，才返回给用户。这样用户在线等待的时间太长，给用户一种卡死了的感觉，这种情况下，用户不能关闭界面，如果关闭了，程序就中断了。

■ 异步

将用户请求放入消息队列，并反馈给用户，程序已经启动，你可以关闭浏览器了。这就是异步。但是用户没有卡死的感觉，会告诉你，你的请求程序已经响应了。你可以关闭界面了。

通俗的解释就是打电话和发短信的区别。

打电话时，对方挂了电话，通信也就中断了，而发短信时，无需考虑对方是否在线，都能进行通信。

■ 概述

Promise是一种用于解决异步问题的思路、方案或者对象方式。

■ 原理

在Promise的内部，有一个状态管理器的存在，有三种状态：pending、fulfilled、rejected。

- Promise 对象初始化状态为 pending。
- 当调用resolve(成功)，会由pending => fulfilled。
- 当调用reject(失败)，会由pending => rejected。

需要记住的是注意promise状态 只能由 pending => fulfilled/rejected, 一旦修改就不能再变（记住，一定要记住，下面会考到）。

当状态为fulfilled（rejected反之）时，then的成功回调函数会被调用，进而进行操作。Promise.then方法每次调用，都返回一个新的Promise对象 所以可以链式写法（无论resolve还是reject都是这样）。

■ 总结

首先，Promise是一个对象，如同其字面意思一样，代表了未来某时间才会知道结果的时间，不受外界因素的干扰。Promise一旦触发，其状态只能变为fulfilled或者rejected，并且已经改变不可逆转。Promise的构造函数接受一个函数作为参数，该参数函数的两个参数分别为resolve和reject，其作用分别是将Promise的状态由pending转化为fulfilled或者rejected，并且将成功或者失败的返回值传递出去。then有两个函数作为Promise状态改变时的回调函数，当Promise状态改变时接受传递来的参数并调用相应的函数。then中的回调的过程为异步操作。catch方法是对then(null,rejectFn)的封装（语法糖），用于指定发生错误时的回调函数。一般来说，建议不要再then中定义rejected状态的回调函数，应该使用catch方法代替。all和race都是竞速函数，all结束的

时间取决于最慢的那个，其作为参数的Promise函数一旦有一个状态为rejected，则总的Promise的状态就为rejected；而race结束的时间取决于最快的那个，一旦最快的那个Promise状态发生改变，那个其总的Promise的状态就变成相应的状态，其余的参数Promise还是会继续进行的。

当然在es7时代，也出现了await/async的异步方案，这会是我们以后谈论的。