fs

阅读: 43229

Node.js内置的fs模块就是文件系统模块，负责读写文件。

和所有其它JavaScript模块不同的是，fs模块同时提供了异步和同步的方法。

回顾一下什么是异步方法。因为JavaScript的单线程模型，执行IO操作时，JavaScript代码无需等待，而是传入回调函数后，继续执行后续JavaScript代码。比如jQuery提供的getJSON()操作：

$.getJSON('http://example.com/ajax', **function** (data) {

console.log('IO结果返回后执行...');

});

console.log('不等待IO结果直接执行后续代码...');

而同步的IO操作则需要等待函数返回：

*// 根据网络耗时，函数将执行几十毫秒~几秒不等:*

**var** data = getJSONSync('http://example.com/ajax');

同步操作的好处是代码简单，缺点是程序将等待IO操作，在等待时间内，无法响应其它任何事件。而异步读取不用等待IO操作，但代码较麻烦。

异步读文件

按照JavaScript的标准，异步读取一个文本文件的代码如下：

'use strict';

**var** fs = **require**('fs');

fs.readFile('sample.txt', 'utf-8', **function** (err, data) {

**if** (err) {

console.log(err);

} **else** {

console.log(data);

}

});

请注意，sample.txt文件必须在当前目录下，且文件编码为utf-8。

异步读取时，传入的回调函数接收两个参数，当正常读取时，err参数为null，data参数为读取到的String。当读取发生错误时，err参数代表一个错误对象，data为undefined。这也是Node.js标准的回调函数：第一个参数代表错误信息，第二个参数代表结果。后面我们还会经常编写这种回调函数。

由于err是否为null就是判断是否出错的标志，所以通常的判断逻辑总是：

**if** (err) {

// 出错了

} **else** {

// 正常

}

如果我们要读取的文件不是文本文件，而是二进制文件，怎么办？

下面的例子演示了如何读取一个图片文件：

'use strict';

**var** fs = **require**('fs');

fs.readFile('sample.png', **function** (err, data) {

**if** (err) {

console.log(err);

} **else** {

console.log(data);

console.log(data.length + ' bytes');

}

});

当读取二进制文件时，不传入文件编码时，回调函数的data参数将返回一个Buffer对象。在Node.js中，Buffer对象就是一个包含零个或任意个字节的数组（注意和Array不同）。

Buffer对象可以和String作转换，例如，把一个Buffer对象转换成String