

Node.js 事件循环

Node.js 是单进程单线程应用程序，但是因为 V8 引擎提供的异步执行回调接口，通过这些接口可以处理大量的并发，所以性能非常高。

Node.js 几乎每一个 API 都是支持回调函数的。

Node.js 基本上所有的事件机制都是用设计模式中观察者模式实现。

Node.js 单线程类似进入一个while(true)的事件循环，直到没有事件观察者退出，每个异步事件都生成一个事件观察者，如果有事件发生就调用该回调函数。

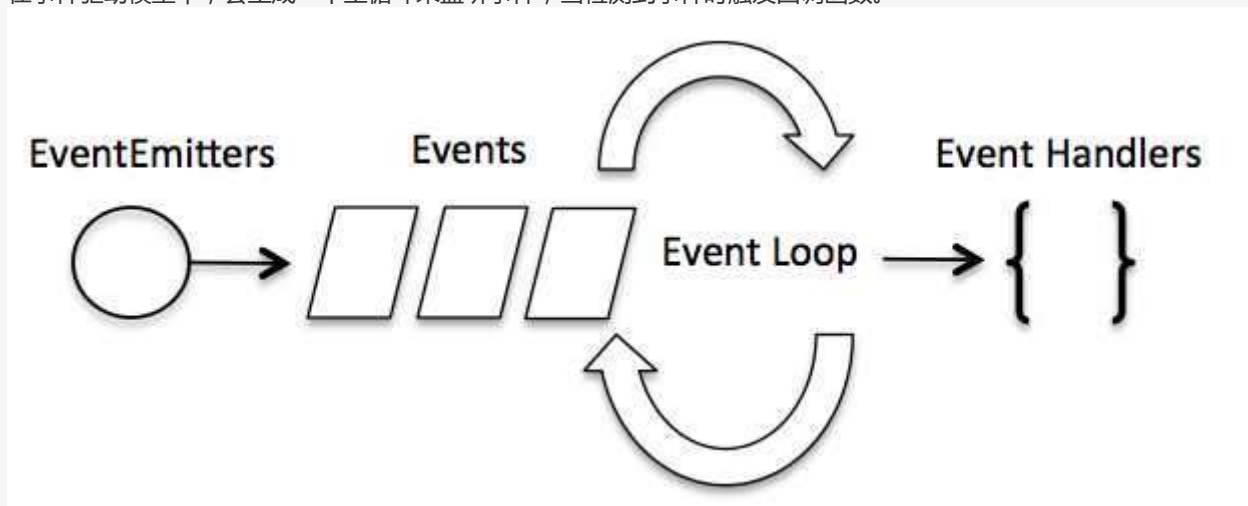
事件驱动程序

Node.js 使用事件驱动模型，当web server接收到请求，就把它关闭然后进行处理，然后去服务下一个web请求。

当这个请求完成，它被放回处理队列，当到达队列开头，这个结果被返回给用户。

这个模型非常高效可扩展性非常强，因为webserver一直接受请求而不等待任何读写操作。（这也被称之为非阻塞式IO或者事件驱动IO）

在事件驱动模型中，会生成一个主循环来监听事件，当检测到事件时触发回调函数。



整个事件驱动的流程就是这么实现的，非常简洁。有点类似于观察者模式，事件相当于一个主题(Subject)，而所有注册到这个事件上的处理函数相当于观察者(Observer)。

Node.js 有多个内置的事件，我们可以通过引入 events 模块，并通过实例化 EventEmitter 类来绑定和监听事件，如下实例：

```
// 引入 events 模块
var events = require('events');
// 创建 EventEmitter 对象
var EventEmitter = new events.EventEmitter();
```

以下程序绑定事件处理程序：

```
// 绑定事件及事件的处理程序
eventEmitter.on('eventName', eventHandler);
```

我们可以通过程序触发事件：

```
// 触发事件
eventEmitter.emit('eventName');
```

实例

创建 main.js 文件，代码如下所示：

```
// 引入 events 模块
var events = require('events');
// 创建 eventEmitter 对象
var eventEmitter = new events.EventEmitter();

// 创建事件处理程序
var connectHandler = function connected() {
  console.log('连接成功。');

  // 触发 data_received 事件
  eventEmitter.emit('data_received');
}

// 绑定 connection 事件处理程序
eventEmitter.on('connection', connectHandler);

// 使用匿名函数绑定 data_received 事件
eventEmitter.on('data_received', function(){
  console.log('数据接收成功。');
});

// 触发 connection 事件
eventEmitter.emit('connection');

console.log("程序执行完毕。");
```

接下来让我们执行以上代码：

```
$ node main.js
连接成功。
数据接收成功。
程序执行完毕。
```

Node 应用程序是如何工作的？

在 Node 应用程序中，执行异步操作的函数将回调函数作为最后一个参数，回调函数接收错误对象作为第一个参数。

接下来让我们来重新看下前面的实例，创建一个 input.txt 文件，内容如下：

菜鸟教程官网地址：www.runoob.com

创建 main.js 文件，代码如下：

```
var fs = require("fs");

fs.readFile('input.txt', function (err, data) {
  if (err){
    console.log(err.stack);
    return;
  }
  console.log(data.toString());
});
console.log("程序执行完毕");
```

以上程序中 fs.readFile() 是异步函数用于读取文件。如果在读取文件过程中发生错误，错误 err 对象就会输出错误信息。

如果没发生错误，readFile 跳过 err 对象的输出，文件内容就通过回调函数输出。

执行以上代码，执行结果如下：

程序执行完毕

菜鸟教程官网地址：www.runoob.com

接下来我们删除 input.txt 文件，执行结果如下所示：

程序执行完毕

Error: ENOENT, open 'input.txt'

因为文件 input.txt 不存在，所以输出了错误信息。

← Node.js 回调函数

Node.js 模块系统 →



3 篇笔记

写笔记