刚刚做了一个非常悲剧的事情，在写文章的中途清理了一遍Cookie，然后没注意一个提交，几个小时写的文章就废了，真是欲哭无泪，所以这篇文章就粗略的回忆一下，肯定没有之前写的心情了。在nodejs中很多对象都会通过事件触发来执行各种操作，例如当一个请求连接的服务器时，net.Server将触发；当一个文件被打开时，fs.readStream将被触发。所有可以触发事件的对象都是events.EventEmitter的一个实例，可以使用require('events')进行引入。在经典的javascript中，事件是通过驼峰来进行命名的，例如：onXXXX，而在nodejs中则不然，可以使用各种字符串，没有特定的限制。而绑定的事件驱动中的方法，英文称为listerners，输出事件处理代理方法，我经常也把他叫做监听器。下面我们来看一下，这个类的一些对象和方法。

**events.EventEmitter类：**

如果需要访问这个类，则需要调用如下方法：

require('events').EventEmitter

如果事件触发器发生了错误，那么将有一个error的事件将被触发，如果没有被重写，默认的处理方法是输出堆栈追踪信息，然后退出应用程序。当然如果添加了一个事件，也将触发一个事件newListener，同样移除的时候也会触发removeListener的事件。那么如何添加和移除呢？请看下面的记录：

**emitter.addListener(event, listener)|emitter.on(event, listener)：**

上面列出的两个方法是一样的，都是添加事件处理器。

server.on('connection', function (stream) {

console.log('someone connected!');

});

**emitter.once(event, listener)：**

上述方法添加的事件将只被执行一次，这个有点像ASP.NET中的Application\_Start一样。

server.once('connection', function (stream) {

console.log('Ah, we have our first user!');

});

**emitter.removeListener(event, listener)：**

移除事件监听程序。

var callback = function(stream) {

console.log('someone connected!');

};

server.on('connection', callback);

// ...

server.removeListener('connection', callback);

*注：当事件被移除时，在其后面的事件队列索引将被改变。*

**emitter.removeAllListeners([event])：**

移除全部事件或指定名称的事件处理器。

**emitter.setMaxListeners(n)：**

设置每个事件的最大绑定数量，默认为10个，如果超过将会发出警告信息，这样有助于防止内存泄露等等。当然不是所有事件都需要这样的限制，可以手动进行设置，如果将其设置为0则表示不限制。

**emitter.listeners(event)：**

返回当前事件的所有监听方法。

server.on('connection', function (stream) {

console.log('someone connected!');

});

console.log(util.inspect(server.listeners('connection'))); // [ [Function] ]

**emitter.emit(event, [arg1], [arg2], [...])**：

队列执行事件。

**EventEmitter.listenerCount(emitter, event)：**

这是一个静态方法，主要用于获取事件处理中的绑定个数。

了解了nodejs事件驱动，将有助于我们下一步的学习，本人学习没有规则，随便选择，所以有可能是错误的学习方法，欢迎指正。