**EventEmitter 类**

下面我们用一个简单的例子说明 EventEmitter 的用法：

//event.js 文件

var EventEmitter = require('events').EventEmitter;

var event = new EventEmitter();

event.on('some\_event', function() {

    console.log('some\_event 事件触发');

});

setTimeout(function() {

    // 发送事件

    event.emit('some\_event');

}, 1000);

EventEmitter 的每个事件由一个事件名和若干个参数组成

让我们以下面的例子解释这个过程：

var events = require('events');

var emitter = new events.EventEmitter();

// 订阅事件 someEvent

emitter.on('someEvent', function(arg1, arg2) {

    console.log('listener1', arg1, arg2);

});

// 订阅事件

emitter.on('someEvent', function(arg1, arg2) {

    console.log('listener2', arg1, arg2);

});

// 触发事件

emitter.emit('someEvent', 'arg1 参数', 'arg2 参数');

执行以上代码，运行的结果如下：

**error 事件**

EventEmitter 定义了一个特殊的事件 error

当 error 被触发时，如果没有响应的监听器，Node.js 会把它当作异常，退出程序并输出错误信息，如下：

var events = require('events');

var emitter = new events.EventEmitter();

emitter.emit('error');

运行时会显示以下错误：

node.js:201

throw e; // process.nextTick error, or 'error' event on first tick

**EventEmitter的使用场景**

我们使用的很多模块如 fs、net、 http 等，他们返回的都是EventEmitter对象