

events事件触发器

- Events模块是Node对“发布/订阅”模式（publish/subscribe）的实现。一个对象通过这个模块，向另一个对象传递消息。
- 该模块通过EventEmitter属性，提供了一个构造函数。该构造函数的实例具有on方法，可以用来监听指定事件，并触发回调函数。
- 任意对象都可以发布指定事件，被EventEmitter实例的on方法监听到。

基本用法

- ① 加载events模块后，通过EventEmitter属性建立了一个EventEmitter对象实例：

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

- 或通过定义扩展类MyEmitter继承EventEmitter，并建立了一个EventEmitter子类对象实例

```
const EventEmitter = require('events');
//继承
class MyEmitter extends EventEmitter {}
//实例化
const myEmitter = new MyEmitter();
```

- ② 注册监听器:通过on方法为eventName事件指定事件处理程序(回调函数eventHandler)：

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

- ③ 通过emit方法触发eventName事件：

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

实例：

- ① 基本使用

```
// 引入 events 模块
const EventEmitter = require('events');
//继承
class MyEmitter extends EventEmitter {}
//实例化
```

```
const myEmitter = new MyEmitter();

//为名为event绑定事件处理程序
myEmitter.on('event', () => {
  console.log('触发事件');
});

//触发名为event的事件
myEmitter.emit('event');
```

- ② 将 `参数` 和 `this` 传给监听器(事件处理程序)

```
// 引入 events 模块
const EventEmitter = require('events');
//继承
class MyEmitter extends EventEmitter {}
//实例化
const myEmitter = new MyEmitter();

myEmitter.on('event', function(a, b) {
  console.log(a, b, this, this === myEmitter);
  //输出
  /*
  aaaa bbbb MyEmitter {
    _events: [Object: null prototype] { event: [Function] },
    _eventsCount: 1,
    _maxListeners: undefined,
    [Symbol(kCapture)]: false
  } true
  */
});

myEmitter.emit('event', 'aaaa', 'bbbb');
```

- 注意：使用 ES6 的箭头函数作为监听器。但 `this` 关键词不会指向 `EventEmitter` 实例

```
//....
const myEmitter = new MyEmitter();
myEmitter.on('event', (a, b) => {
  console.log(a, b, this);
  // 打印: a b {}
});
myEmitter.emit('event', 'a', 'b');
```

- ③ 仅处理事件一次
- 使用 `eventEmitter.once()` 可以注册最多可调用一次的监听器。当事件被触发时，监听器会被注销，然后再调用。

```
// 引入 events 模块
const EventEmitter = require('events');
//继承
class MyEmitter extends EventEmitter {}
//实例化
const myEmitter = new MyEmitter();

let m = 0;
myEmitter.once('event', () => {
  console.log(++m);
});
myEmitter.emit('event');
// 打印: 1
myEmitter.emit('event');
// 不触发
```

- ④ 其他事件应用具体详见手册说明。