

*the*  
**NODE FIRM**

Copyright© 2013 The Node Firm. All Rights Reserved.

# EVENT EMITTER

## WHAT ARE EVENTEMITTERS?

A way to emit state changes and messages to subscribers

```
var EventEmitter = require('events').EventEmitter;  
  
var emitter = new EventEmitter();
```

## BUILT IN EVENTEMITTERS

EventEmitters are heavily relied on in node core:

- http
- net
- filesystem
- domains
- and more!

## ADDING LISTENERS

Add listeners using the `on()` function:

### 01\_on.js

```
var EventEmitter = require('events').EventEmitter;
var emitter = new EventEmitter();

// register interest in 'ping' events
emitter.on('ping', function firstPingListener() {
  console.log('first listener: ping');
});

// another 'ping' event listener
emitter.on('ping', function secondPinglistener() {
  console.log('second listener: ping');
});

// trigger a 'ping' event
emitter.emit('ping');
emitter.emit('ping');
```

## ADDING ONE-SHOT LISTENERS

### 02\_once.js

```
var EventEmitter = require('events').EventEmitter;
var emitter = new EventEmitter();

emitter.once('ping', function() {
  console.log('received ping'); // only called once
});

emitter.emit('ping');
emitter.emit('ping');
```

## REMOVING A LISTENER

```
// remove a single listener
emitter.removeListener('ping', pingListener1);

// remove all listeners
emitter.removeAllListeners('ping');
```

# EXAMPLE

## 03\_http\_server.js:

```
var server = require('http').createServer();
var closing = false;
var port = process.argv[2] || 9000;

server.on('request', function(req, res) {
  res.setHeader('Connection', 'close'); // turn off keep-alive
  res.end('Hello World!');
  if (!closing) {
    closing = true;
    server.close();
  }
});

server.once('listening', function() {
  console.log('Server listening on port %d', port);
});

server.once('close', function() {
  console.log('Server is closing. Bye bye!');
});

server.listen(port);
```



## SENDING MESSAGES

emit alternating "tic" and "tac" events every second

**04\_tic\_tac\_emitter.js:**

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

var emitter = new EventEmitter;

var tick = true;
var count = 0;

setInterval(function() {
  count++;
  if (tick) emitter.emit('tic', count);
  else emitter.emit('tac', count);
  tick = !tick;
}, 1000);

emitter.on('tic', function(count) {
  console.log(['%d] TIC', count);
});

emitter.on('tac', function(count) {
  console.log(['%d] TAC', count);
});
```

## SUMMARY

- EventEmitter is a core Node.js construct
- An event has an event name, identified by a string
- Add event listener: `on(name, callback)`
- Add an one time listener: `once(name, callback)`
- Remove a listener: `removeListener(name, fn)`
- Remove all listeners: `removeAllListeners(name)`
- Emit an event: `emit(name [, data, ...])`