

原生 nodejs 结合 Socket.io 实现服务器和客户端的相互通信

主讲教师：（大地）

合作网站：www.itying.com （IT 营）

目录

- 1、 安装 Socket.io.....1
- 2、 写原生的 JS，搭建一个服务器，server 创建好之后，创建一个 io 对象。..... 1
- 3、 服务器端通过 emit 广播，通过 on 接收广播..... 2
- 4、 客户端端通过 emit 广播，通过 on 接收广播..... 3

1、 安装 Socket.io

网址：<http://socket.io/>

```
npm install socket.io
```

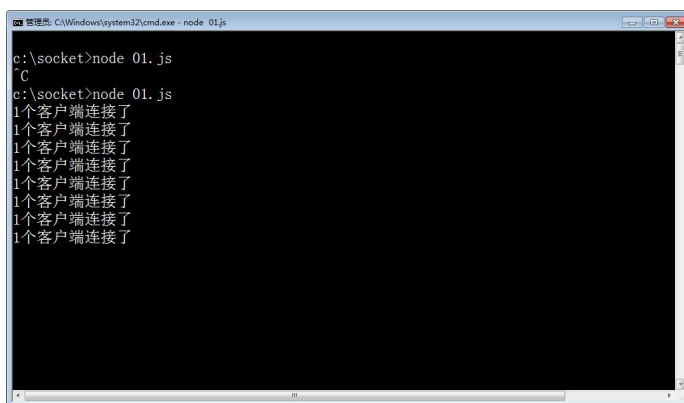
2、写原生的 JS，搭建一个服务器，server 创建好之后，创建一个 io 对象。

```
var http = require("http");
var server = http.createServer(function(req,res){
    if(req.url == "/"){
        //显示首页
        fs.readFile("./index.html",function(err,data){
            res.end(data);
        });
    }
});
var io = require('socket.io')(server);
//监听连接事件
io.on("connection",function(socket){
    console.log("1 个客户端连接了");
})
server.listen(3000,"127.0.0.1");
```

写完这句话之后，你就会发现，`http://127.0.0.1:3000/socket.io/socket.io.js` 就是一个 js 文件的地址了。

现在需要制作一个 index 页面，这个页面中，必须引用秘密 js 文件。调用 io 函数，取得 socket 对象。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
  <h1>我是 index 页面，我引用了秘密 script 文件</h1>
  <script type="text/javascript" src="/socket.io/socket.io.js"></script>
  <script type="text/javascript">
    var socket = io();
  </script>
</body>
</html>
```



至此，服务器和客户端都有 socket 对象了。服务器的 socket 对象：

3、服务器端通过 emit 广播，通过 on 接收广播

```
var io=require('socket.io')(server);
io.on('connection',function(socket){
  console.log('和服务器建立连接了');
  socket.on('to-server',function(data){
    console.log('客户端说:'+data);
```

```

        socket.emit('to-client', '我是服务器返回的数据');
    })
    socket.on('disconnect', function() {
        console.log('断开连接了');
    })
})

```

每一个连接上来的用户，都有一个 socket。由于我们的 emit 语句，是 socket.emit() 发出的，所以指的是向这个客户端发出语句。

广播，就是给所有当前连接的用户发送信息：

```

var io=require('socket.io')(server);
io.on('connection', function(socket) {
    console.log('和服务器建立了连接')
    socket.on('to-server', function(data) {
        console.log('客户端说:' + data);
        io.emit('to-client', '我是服务器返回的数据');
    })
    socket.on('disconnect', function() {
        console.log('断开连接了');
    })
})

```

4、客户端通过 emit 广播，通过 on 接收广播

```

var socket=io.connect('http://localhost:3001');
socket.on('connect', function() {
    console.log('客户端和服务端建立了连接');
})
socket.on('disconnect', function() {
    console.log('客户端和服务端断开连接了');
})
socket.on('to-client', function(data) {
    console.log('客户端说:' + data);
})
var btn=document.getElementById('button');
btn.onclick=function() {
    socket.emit('to-server', '我是客户端的数据');
}

```

