基础入门

Node.js 简介

服务器端的javascript

Node repl

交互式命令解析器，检验js和node.js

直接交互

Node filename.js 通过文件运行

Node的基本命令

.help 显示所有的基本命令

.clear 清除当前运行内容

.exit 退出node

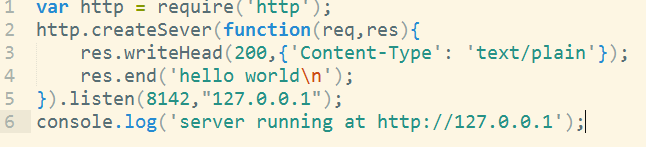
Node 会智能显示对象，不会冗长，刷屏

Node集成了大量的应用库，可以直接调用(包含其他库)

Require(‘http’); 调用http模块

Node 与php不同在于node本身就是服务器，不需要再别的服务器下运行

第一步是创建服务器



http变量就是新创建的服务器

Listen（）表示监听的端口

createSever(function)工厂模式创建服务器，function表示服务器进行request事件处理的函数，每有一个request，就调用该函数。这种方法称为回调

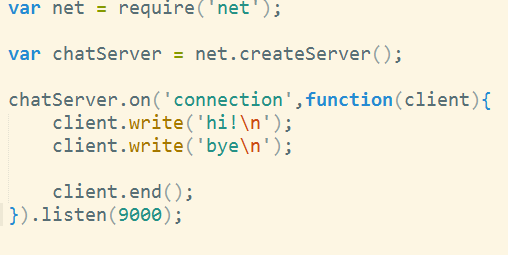
req表示请求对象,res表示响应对象

res.writeHead设置http响应头，没有响应头不能发内容给客户端。状态码200表示ok

res.end(‘string’)表示发送string给客户端后关闭该链接。Res.end()关闭连接

console.log()在标准stdout上显示信息

编写有趣应用



创建TCP服务器

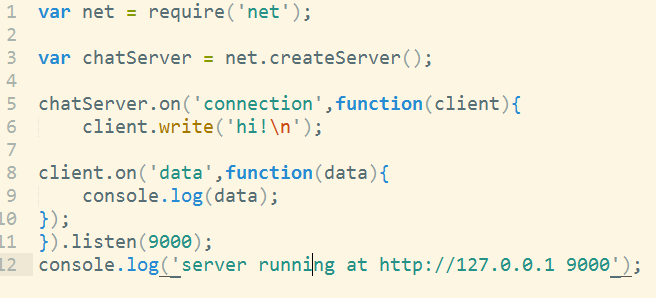
Require（’net’） 表示添加TCP模块

On（）添加事件听器，每当有客户端接入服务器，出发connection事件，并调用function，触发connection事件时传入客户端对应TCP socket对象的引用，我们把这个引用命名为client。

调用client.write()就可以发信息给客户端

Client.end()表示关闭与客户端的链接

用putty连接node创建的服务器，输入localhost或者127.0.0.1，以及listen()的端口号，选择telnet即可访问



在connection回调中再添加一个client.on(‘data’,function(data))。表示关心连接到服务器客户端的data事件。每当客户端发送数据到服务器，该.on()中的function触发。

删掉client.end().

为了继续接受客户端的消息

Javascript不会处理二进制，node自带一个buffer库保存客户端传来数据的二进制格式（还不知道用什么编码）。屏幕上显示的是十六进制字节数据。

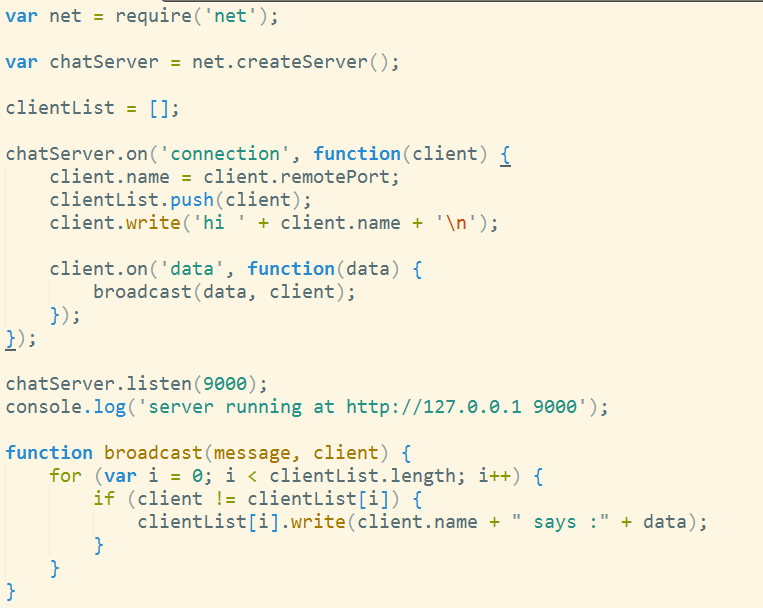
TCP和telnet都可以处理二进制数据

Ctrl+c 可以kill掉当前端口



多个客户端相互通信，创建一个clientList数组保存与服务器通信的客户端，利用列表实现相互通信

服务器相当于信息中转站



通过添加client.name区分 是谁发的消息

闭包绑定了每个client对象和相应的请求

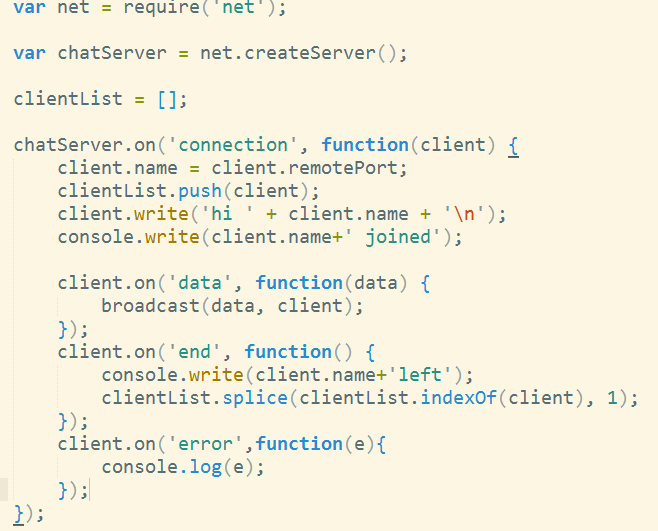
Client.remoteAddress表示客户端所在的IP

Client.remotePort表示客户接受数据的TCP端口

IP可以相同,Port肯定不同

这种情况下，如果一个客户端离开，程序会向断开的客户端写入数据，出现错误

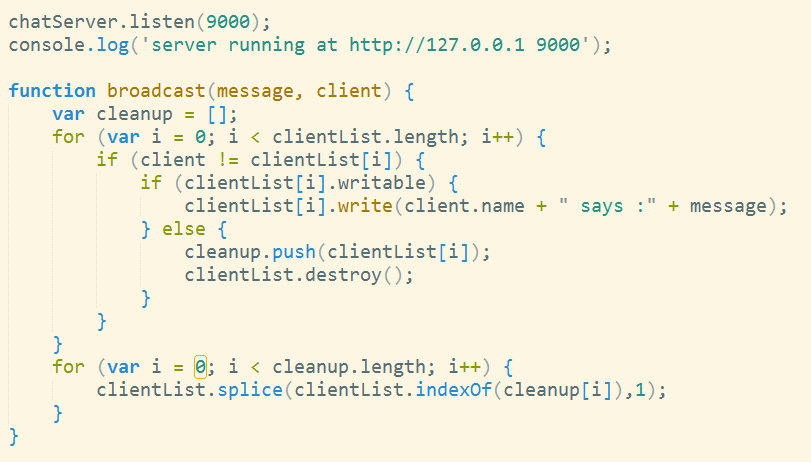
解决客户端离开后，向空socket对象写入或读取数据引发错误



添加纪录，用户连接，用户离开，出现错误

添加client.on(‘end’,function(){…});

当用户断开连接时触发，从列表中删除该用户



调用broadcast先检查用户端口是否可写入（.writable属性）,如果不行，压入cleanup列表，clientList.destory() 将其关闭。

不在clientList循环中移除socket，防止出现未知错误，循环结束后，从clientList中移除不可写的socket