[测试Node.js 2](#_Toc375160634)

[框架选择 2](#_Toc375160635)

[Jasmine-node 2](#_Toc375160636)

[Mocha 2](#_Toc375160637)

[Mock框架 2](#_Toc375160638)

[策略 3](#_Toc375160639)

[问题及解决方案 3](#_Toc375160640)

[1. 重复启动服务器会报错 3](#_Toc375160641)

[测试Socket.IO 5](#_Toc375160642)

[客户端测试 5](#_Toc375160643)

[服务器端测试 5](#_Toc375160644)

# 测试Node.js

## 框架选择

* Jasmine-node
* mocha加上should.js断言库

## Jasmine-node

单元测试使用jasmine-node。在源代码根目录运行jasmine-node spec可运行所有测试。

支持异步测试。

当然也可以进行自动的测试，也可以增加 --autotest选项，当文件发生变化的时候进行自动测试 jasmine-node --autotest  testspec.js。 。启动一次 会自动监控文件的变化，也可以通过 --watch 路径 。来检测变化的文件

## Mocha

支持在浏览器中以Html页面显示测试结果。

异步测试与jasmine-node相似（都是调用done()）

有before, after, before each, after each钩子，而jasmine只有before each, after each钩子

。。。。。。

## Mock框架

暂时不考虑。

貌似SinonJs可以用在node.js中

MaryJane可以用

## 策略

优先使用Jasmine-node

然后再试用Mocha+should.js

对两者进行比较，决定最终选择

## 问题及解决方案

### 重复启动服务器会报错

**问题描述**

启动服务器时（如node server.js），提示一下错误：

events.js:71

throw arguments[1]; // Unhandled 'error' event

^

Error: listen EADDRINUSE

**原因分析**

遇到这种情况，就说明当前监听的端口已经被占用

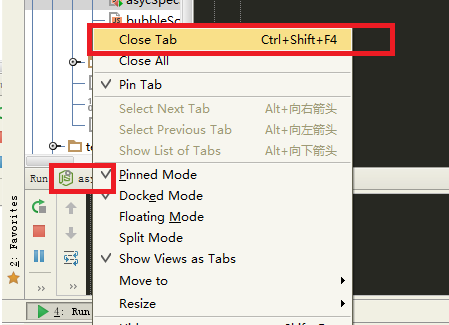
**解决方案**

1. 杀掉node进程，重启服务器

“杀掉node进程”有两种方法：

一是在cmd中杀掉进程：在win7的cmd命名中输入：tskill node

二是直接在webstorm中关掉服务器进程：



重启服务器：

在webstorm中run node.js服务器文件（如server.js）

1. 更改监听端口

更改服务器端口，然后再启动服务器（如更改server.js监听的端口号，然后再run server.js）

1. 使用nodemon运行js

Nodemon是一个对node.js的包装器，它貌似可以重启服务器。

如在cmd命令中输入：nodeme server.js

则即使服务器已经打开，也可以重启服务器（如果用node启动，则会报错！）

Nodemon参考资料：<https://github.com/remy/nodemon>

# 测试Socket.IO

## 客户端测试

使用测试替身来测试

在客户端使用jasmine来测试

## 服务器端测试

* 可以在客户端使用jasmine，通过调用Socket.IO客户端API来与服务器通信，从而测试Socket.IO搭建的服务器端。
* 安装socket.io-client模块后，也可以在服务器端测试。

使用“npm install socket.io-client”命令安装socket.io-client模块后，即可以在服务器端运行socket.io的客户端API，发起客户端请求。然后通过验证Socket.IO服务器端的回应信息，来进行测试。