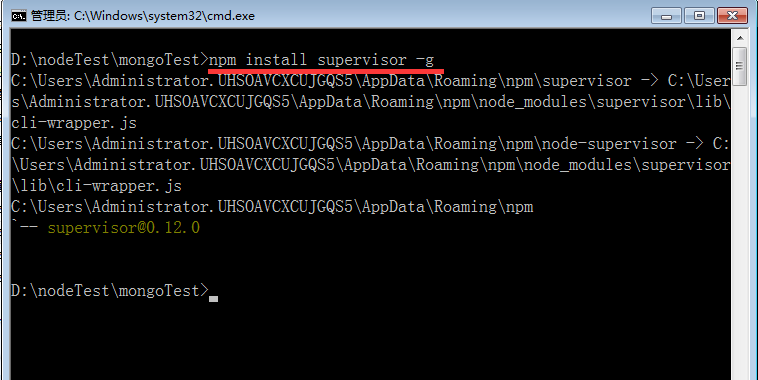
这个模块的使用很方便，只需两步，1、安装。2、监听变化

打开命令行 全局安装   npm install supervisor -g



由于是全局安装，所以命令行在哪个文件路径运行都不影响，我这里在 mongoTest文件夹运行

我现在自己的某个文件夹中创建demo.js。写入如下代码：这是我们前面写过的创建http服务的代码，这里直接复制即可。

[复制代码](javascript:void(0);)

var http = require("http");

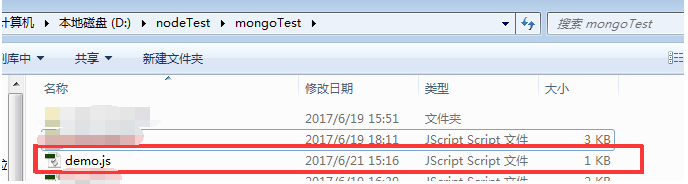
http.createServer(function(request,response){

response.end("haha");//表示请求结束，将结果返回给浏览器

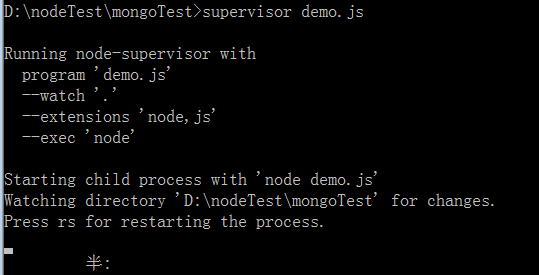
}).listen(3000);

[复制代码](javascript:void(0);)

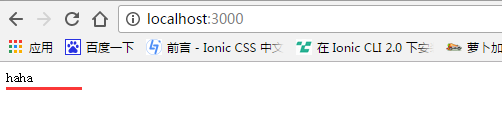
下图表示，我将 demo.js创建在了mongoTest文件夹中。（文件夹中的其他文件忽略）



 然后运行 demo.js ，这里的运行就不再用 node demo.js了， 而是使用 supervisor demo.js



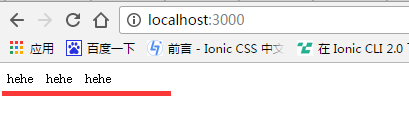
运行起来之后，就可以通过浏览器访问



现在我们来改变下demo.js中输出的代码

将 response.end("haha") 中输出的内容该变，比如改为  response.end("hehe  hehe  hehe")

然后直接刷新页面，就可以看到修改的结果被输出了（以前我们需要重新运行 node demo.js才可以）



好啦，自动重启服务就搞定了

 以后要使用就不用再安装了，因为已经安装在全局了。另外最好只对创建了http服务的node文件才用supervisor运行，如果一般的文件这样运行，会不断的重启，很耗费内存。

upervisor的其他配置如下，有兴趣的可以研究研究

[复制代码](javascript:void(0);)

//要监控的文件夹或js文件，默认为'.'

-w|--watch <watchItems>

//要忽略监控的文件夹或js文件

-i|--ignore <ignoreItems>

//监控文件变化的时间间隔（周期），默认为Node.js内置的时间

-p|--poll-interval <milliseconds>

//要监控的文件扩展名，默认为'node|js'

-e|--extensions <extensions>

//要执行的主应用程序，默认为'node'

-x|--exec <executable>

//开启debug模式（用--debug flag来启动node）

--debug

//安静模式，不显示DEBUG信息

-q|--quiet

[复制代码](javascript:void(0);)