

如果我们使用PHP来编写后端的代码时,需要Apache或者Nginx的HTTP服务器,并配上mod_php5模块和php-cgi.

从这个角度看,整个"接收HTTP请求并提供Web页面"的需求根本不需要PHP来处理,不过对Node.js来说,概念完全不一样了.使用Node.js时,我们不仅仅在实现一个应用,同时还实现了整个HTTP服务器.事实上,我们的Web应用以及对应的Web服务器基本上是一样的.

Node.js应用的组成:

1. 引入required模块:可以使用require指令来载入Node.js模块
2. 创建服务器:服务器可以监听客户端的请求,类似与Apache,Nginx等Http服务器.
3. 接收请求与响应请求:服务器很容易创建,客户端可以使用浏览器或中断发送HTTP请求,服务器接收请求后返回响应数据

创建Node.js应用

步骤一.引入required模块

我们使用require指令来载入http模块,并将实例化的Http赋值给变量http,如:

```
var http = require("http")
```

步骤二.创建服务器

我们使用http.createServer()方法创建服务器,并使用listen方法绑定8888端口.函数通过request,response参数来接收和响应数据

server.js

```
var http = require('http');
```

```
http.createServer(function (request, response) {
```

```
// 发送 HTTP 头部
```

```
// HTTP 状态值: 200 : OK
// 内容类型: text/plain
response.writeHead(200, {'Content-Type': 'text/plain'});

// 发送响应数据 "Hello World"
response.end('Hello World\n');
}).listen(8888);

// 终端打印如下信息
console.log('Server running at http://127.0.0.1:8888/');
```

以上代码完成了一个可以工作的HTTP服务器。

阻塞是按顺序执行的，而非阻塞是不需要按顺序的，所以如果需要处理回调函数的参数，我们就需要写在回调函数内。

