TURING 图灵程序设计丛书

[PACKT] PUBLISHING

[美] David Herron 著　　鄢学鹃 吴天豪 廖健 译

Node Web Development

Node Web开发

人民邮电出版社
POSTS & TELECOM PRESS

[美] David Herron 著　　郭学鸣 吴天泉 邵健 译

Node Web Development

Node Web开发

人民邮电出版社
北京

图书在版编目（CIP）数据

Node Web开发 / （美）赫伦（Herron,D.）著 ; 鄢学鹃, 吴天豪, 廖健译. -- 北京 : 人民邮电出版社, 2012.4
（图灵程序设计丛书）
书名原文: Node Web Development
ISBN 978-7-115-27832-6

Ⅰ. ①N… Ⅱ. ①赫… ②鄢… ③吴… ④廖… Ⅲ. ①网页制作工具－程序设计 Ⅳ. ①TP393.092

中国版本图书馆CIP数据核字(2012)第056420号

内 容 提 要

Node是一个服务器端的JavaScript解释器，是构建快速响应、高度可扩展网络应用的轻量高效的平台。Node使用事件驱动和非阻塞的I/O模型，非常适合数据密集、对实时响应要求高的分布式应用。微软、eBay、LinkedIn、雅虎等世界知名公司及网站均有使用Node的成功案例。

本书是基于Node开发Web应用的实用指南，全书共分6章，通过示例详尽介绍了Node的背景、原理及应用方法。全书内容涉及Node简介、Node安装、Node模块、实现不同版本的简单应用、实现简单的Web服务器和EventEmitter，以及数据存储和检索。另外，本书涵盖了Node服务器端开发的主要挑战及应对方案。

本书适合Web前、后端开发人员学习参考。

图灵程序设计丛书

Node Web开发

◆ 著 [美] David Herron
译 鄢学鹃 吴天豪 廖 健
责任编辑 毛倩倩

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：7.25
字数：176千字 2012年4月第1版
印数：1－5 000册 2012年4月北京第1次印刷

著作权合同登记号 图字：01-2012-1950号

ISBN 978-7-115-27832-6

定价：35.00元

读者服务热线：(010)51095186转604 印装质量热线：(010)67129223

反盗版热线：(010)67171154

站在巨人的肩上

Standing on Shoulders of Giants

www.ituring.com.cn

版权声明

审稿人介绍

Blagovest Dachev 2002年起开始编写Web软件，从事过各种各样的开发工作，从HTML、CSS、JavaScript开发开始，后转站服务器和数据库领域。他很早就加入了Node.js开发行列，并已经为多个开源项目做出贡献，现在是Dow Jones &Company的一名软件工程师，致力于开发小工具框架，为第三方提供在自己网站上搜索和展示新闻的功能。

他曾就读于马萨诸塞大学安姆斯特分校，并在那儿参与了信息检索方面的研究，两次参与“谷歌编程之夏”（Google Summer of Code）活动，且与他人合著了几篇论文。

我想感谢我的母亲Tatiana，谢谢她给我的爱，对我的无私奉献，还有多年来一直激励我前进的力量；感谢父亲给了我一个幸福快乐的童年。

Matt Ranney 他很早就采用并参与开发Node.js，且是Voxer（使用Node作为后台服务器）的创建者之一。

致　　谢

我想感谢很多人。

感谢母亲Evelyn为我做的一切，感谢父亲Jim、姐姐Patti、哥哥Ken。如果没有你们，真不敢想象我的生活会是个什么样子。

感谢女友Maggie一直以来对我的陪伴，给我的鼓励和信赖，她聪明睿智，总能在适当的时候为我打气。希望我们能永远在一起。

感谢肯塔基大学的Ken Kubota博士对我的信任，是他给了我第一份与计算机相关的工作。6年来，我不仅仅学到了计算机系统维护技巧，还学到了很多其他知识。

感谢我的前老板，感谢肯塔基大学数学科学系、Wollongong集团、MainSoft、VXtreme、Sun Microsystems和雅虎公司，还有每一位曾和我一起工作的同事。非常感谢我的前任经理Tina Su，是她不断鼓励我公开演讲和撰写文章，这对于一个内向的软件工程师而言绝非易事。非常感谢雅虎给我机会参与其内部的Node.js项目开发，并能够照顾我写作本书的需要。

感谢Packt Publishing给我写书的机会和给予的专业指导，让我认识到写书是我的梦想。

感谢Ryan Dahl、Isaac Schlueter和其他Node核心团队成员，是他们的智慧和眼光造就了如此有趣易用的软件开发平台。在诸多同类平台中，Node独树一帜，它既强大又好用，实现这么棒的平台需要极宽广的眼界。

www.packtpub.com

支持文件、电子书及打折优惠

欢迎读者访问www.packtpub.com下载支持文件及其他相关资源。

Packt为所有已出版图书提供电子版（PDF和ePub文件）。你可以在www.packtpub.com上升级电子书，且纸质书的用户有权享受购买其电子版的折扣。要了解更多信息，请发邮件到service@packtpub.com。

在www.packtpub.com还可以阅读免费的技术文章，通过注册收取一系列最新促销信息，并获得Packt纸质书及电子书的独家折扣或享受特价购书。

http://PacktLib.PacktPub.com

你是否经常受到一些IT难题的困扰？PacktLib是Packt的在线数字图书馆。在这里，你可以访问、阅读和搜索整个Packt图书馆里的图书。

为什么订阅

- 可以搜索Packt出版的所有图书；
- 可以复制、粘贴、打印内容并在内容上添加书签；
- 可以通过浏览器实现按需访问。

Packt 注册用户大礼

在www.packtpub.com上注册一个Packt账号，就可以立即访问PacktLib并自由浏览9本图书。注册 → 登录 → 浏览，就这么简单。

前　言

欢迎光临Node（也叫Node.js）开发的世界。Node是一种新兴的软件开发平台，它将JavaScript从Web浏览器移植到常规的服务器端。Node运行在Chrome的高速V8引擎上，并附带了一个快速、健壮的异步网络I/O组件库。Node主要用于构建高性能、高可扩展的服务器和客户端应用，以实现真正“实时的Web应用”。

在经过数年尝试用Ruby和其他语言实现Web服务器组件之后，Ryan Dahl在2009年开发了Node平台。这个探索使他从使用传统的、基于线程的并发模型转向使用事件驱动的异步系统，因为后者更简单（多线程系统以难于开发著称），系统开销更低（与对每个连接维护一个线程相比），因而能提高相应的速度。Node旨在提供一个“创建可扩展网络服务器的简单方式”。这个设计受到了Event Machine（Ruby）和Twisted框架（Python）的影响，并和它们有些类似。

本书致力于讲述如何用Node构建Web应用。我们会在书中介绍快速学习Node时一些必需的重要概念。本书会教你编写真正的应用，剖析其工作原理，并讨论如何在程序中应用这些理念。我们需要安装Node和npm，学习安装和开发npm包及Node模块。此外，我们还会开发一些应用，度量长时间运行的计算在Node的事件循环中的响应能力，介绍将高负载的工作分派到多个服务器的方法，并介绍Express框架。

本书内容

第1章“Node入门”，介绍了Node平台。这一章讲述了Node的用途、技术构架上的选择、Node的历史和服务器端JavaScript的历史，然后介绍为什么JavaScript仍将受困于浏览器。

第2章“安装并配置Node”，介绍如何配置Node开发环境，包括多种从源码编译和安装的场景，还会简单介绍在开发环境中如何部署Node。

第3章“Node模块”，解释了作为开发Node应用基本单位的模块。我们会全面介绍并开发Node模块。然后进一步介绍Node包管理器npm，给出一些使用npm管理已安装包的例子，还将涉及开发npm包并将其发布出来供他人使用。

第4章“几种典型的简单应用”，在读者已经有一些Node基础知识后，开始探索Node应用的开发。我们会分别使用Node、Connect中间件框架和Express应用框架开发一个简单的应用。虽然应用比较简单，但是我们可以通过其开发探索Node的事件循环，处理长时间的运算，了解异步和同步算法以及如何将繁重的计算交给后台服务器。

第5章“简单的Web服务器、EventEmitter和HTTP客户端”，介绍了Node里的HTTP客户端和服务器对象。我们会在开发HTTP服务器和客户端应用的同时全面深入介绍HTTP会话。

第6章“存取数据”，探讨大部分应用都需要的长期可靠的数据存储机制。我们会用SQL和MongoDB数据库引擎实现一个应用。在此期间，我们将用Express框架实现用户验证，更好地展示出错页面。

阅读要求

目前，我们一般会采用源码的方式安装Node，这种方式可以很好地用在类Unix和符合POSIX标准的系统上。当然，在接触Node之前，谦逊的心态是必需的，但最为重要的事情还是让大脑供血充足。

从源码安装的方式需要一个类Unix或类POSIX系统（比如Linux、Mac、FreeBSD、OpenSolaris等）、新的C/C++编译器、OpenSSL库和Python 2.4或更新版本。

Node程序可以用任何文本编辑器来写，不过一个能处理JavaScript、HTML、CSS等的文本编辑器会更有帮助。

尽管本书介绍的是Web应用开发，但你并不需要拥有一个Web服务器。Node有自己的Web服务器套件。

读者对象

本书写给所有想在一个新的软件平台上开拓新编程模式的软件工程师。

服务器端程序员或许能看到一些新奇的概念，对Web应用开发产生新的理解。JavaScript是一门强大的语言，Node的异步特性发挥了JavaScript的优势。

浏览器端JavaScript“攻城师”或许会觉得在Node中使用JavaScript和编写与DOM操作无关的JavaScript代码很有趣。（Node平台上没有浏览器，所以也没有DOM，除非你安装JSDom。）

虽然本书各章内容由浅入深，循序渐进，但到底如何阅读本书悉听尊便。

本书需要读者知道如何编写软件，并且对JavaScript等编程语言有所了解。

排版约定

在本书中，读者会发现不同的文本样式。下面是这些样式的示例和说明。

正文中的代码使用特殊字体：“`http`对象封装HTTP协议，它的`http.createServer`方法会创建一个完整的Web服务器，而`.listen`方法用于监听特定的端口。”

代码块是这样的：

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
```

```
}).listen(8124, "127.0.0.1");
console.log('Server running at http://127.0.0.1:8124/');
```

代码块中会加粗突出显示代码，这表示需要读者格外注意：

```
var util = require('util');
var A = "a different value A";
var B = "a different value B";
var m1 = require('./module1');
util.log('A='+A+' B='+B+' values='+util.inspect(m1.values()));
```

命令行的输入输出是这样的：

```
$ sudo /usr/sbin/update-rc.d node defaults
```

新术语及重要词汇都会加粗显示。你将在屏幕上看到的文字，比如菜单或对话框中的文字，会这样在正文中提到："一个真正安全的系统至少会有用户名和密码输入框。不过，我们这里就直接让用户单击Login按钮了。"

读者反馈

我们始终欢迎来自读者的反馈意见。我们想知道读者对本书的看法，读者喜欢哪些内容或不喜欢哪些内容。读者真正深有感触的反馈，对于我们开发图书产品至关重要。

一般的反馈可以发邮件到feedback@packtpub.com，但请在邮件标题中注明相关书名。

如果有关于新书的建议，你可以登录www.packtpub.com，填写SUGGEST A TITLE表单或者向suggest@packtpub.com发送邮件。

如果你在某个领域积累了丰富的经验，想写一本书，或者愿意与人合著或审校某本书，请阅读www.packtpub.com/authors上的作者指南。

读者服务

现在你已是Packt引以为荣的读者了，因此我们特别要交待几件事，以保障你作为读者的最大权益。

下载示例代码

在www.packtpub.com通过自己的账号购买图书的读者，可以下载所有已购买图书的代码[①]。如果这本书是你在其他地方购买的，访问www.packtpub.com/support并注册，我们将通过电子邮件将相关文件发送给你。

① 本书的代码文件也可在图灵社区（ituring. com. cn）本书网页免费注册下载。——编者注

勘误

虽然我们会全力确保本书内容的准确性，但错误仍在所难免。如果你发现了本书中的错误（包括文字和代码错误），而且愿意向我们提交这些错误，我们会十分感激。这样一来，不仅可以减少其他读者的疑虑，也有助于本书后续版本的改进。要提交错误，请访问www.packtpub.com/support，选择相关图书，单击errata submission form链接，然后输入勘误信息。经过验证后，你提交的勘误信息就会添加到已有的勘误列表中。要查看已有的勘误信息，请访问www.packtpub.com/support并选择相关图书。

反盗版声明

网上各种形式的盗版是一直存在的问题。Packt非常重视版权和许可证的保护。如果你在网上遇到以任何形式非法复制的我方作品，请尽快告知我们相关的地址或网站名称，以便我们采取补救措施。

请把邮件发送到copyright@packtpub.com，并在邮件里注明涉嫌侵权资料的链接。

感谢你帮助我们保护作者和我们为你带来有价值内容的能力。

疑难解答

如果对本书的某些方面有疑问，请将电子邮件发送到questions@packtpub.com，我们会尽力解决。

目　录

第1章 Node入门

无论开发Web应用、应用服务器、任何类型的网络服务器或客户端还是通用编程，Node都是一个令人兴奋的新平台。它把异步I/O和服务器端JavaScript巧妙地组合在一起，聪明地运用了强大的JavaScript匿名函数和单线程执行的事件驱动架构，是专为网络应用的极高扩展性而设计的。

Node模型与大规模使用线程的普通应用服务器平台截然不同。由于使用事件驱动架构，内存占用量低，吞吐量高，且编程模型更简单。Node平台正处于高速成长阶段，很多人使用Node就是为了替代传统方法（使用Apache、PHP、Python等）。

Node的核心是一个独立的JavaScript虚拟机，通过扩展之后可适用于通用编程，并明确地专注于应用服务器开发。Node平台和开发Web应用的常用编程语言（PHP、Python、Ruby、Java等）不是一类东西，与那些向客户端提供HTTP协议的容器（Apache、Tomcat、Glassfish等）也没有直接可比性。而且，很多人认为它非常有可能取代传统的Web应用开发相关技术。

Node实现以非阻塞的I/O事件循环机制和文件与网络I/O库为中心，一切都以（来自Chrome浏览器的）V8 JavaScript引擎为基础。这个I/O库足以实现采用任何TCP或UDP协议的、任意类型的服务器，无论是DNS服务器，还是采用HTTP、IRC、FTP服务器。虽然它支持针对任何网络协议开发服务器或客户端，但最常用于构建普通网站，这种情况下可以取代像Apache/PHP或Rails这样的框架。

本书将向你介绍Node。我们假定你已知道如何编写软件、熟悉JavaScript，且对如何用其他语言开发Web应用有所了解。我们将以开发实际应用为例，因为阅读实际的代码是最好的学习方式。

1.1 Node能做什么

Node是脱离浏览器编写JavaScript应用的平台。这里的JavaScript并非我们在浏览器中熟悉的JavaScript，Node没有内置DOM，也没有任何浏览器的功能。但它使用JavaScript语言和异步I/O框架，是一个强大的应用开发平台。

Node无法用于实现桌面GUI应用。目前Node既没有内置相当于Swing（或SWT）的功能组

件[①]，也没有Node扩展GUI工具包，而且不能内嵌到Web浏览器中。如果有可用的GUI工具包，Node就能用于构建桌面应用。一些项目已经开始为Node创建GTK绑定[②]，它能提供一个跨平台的GUI工具包。Node使用的V8引擎带有一个扩展API，允许编入C/C++代码以扩展JavaScript或集成原生代码库。

除了能原生执行JavaScript外，Node捆绑的模块还能提供如下功能：

- 命令行工具（shell脚本风格）；
- 交互式TTY风格编程（REPL[③]）；
- 出色的进程控制函数能监控子进程；
- 用Buffer对象处理二进制数据；
- 使用全面事件驱动回调函数的TCP或UDP套接字；
- DNS查找；
- 基于TCP库的HTTP和HTTPS客户端/服务器；
- 文件系统的存取；
- 内置了基于断言的单元测试能力。

Node的网络层是底层，但易于使用。例如，HTTP模块可以让你仅用几行代码编写出一个HTTP服务器（客户端），但这层让程序员非常贴近协议的要求，且让你精确控制将返回的请求响应的HTTP头。PHP程序员通常不需要关心HTTP头，但Node程序员需要。

换句话说，用Node编写HTTP服务器非常容易，但通常Web应用开发者不需要在这种细节层面上费神。例如，由于Apache已经存在，PHP程序员就不需要实现其HTTP服务器部分了。Node社区已经开发出许多类似于Connect的Web应用框架，让开发者能够快速配置可提供所有基础内容（会话、cookie、静态文件和日志等）的HTTP服务器，从而把时间和精力都放在业务逻辑上。

服务器端 JavaScript

有点不耐烦了？你正一边摇头一边抱怨："在服务器上用一门浏览器语言做什么？"实际上，JavaScript在浏览器之外有着悠久且大量鲜为人知的历史。JavaScript就像其他语言一样是一门编程语言，而你的问题可能应该这样："为什么JavaScript会一直被困在浏览器中?"

---

① Swing是一个为Java设计的工具包，是Java基础类的一部分，具体见http://zh.wikipedia.org/wiki/Swing_(Java)。SWT（Standard Widget Toolkit）最初由IBM开发，是一套用于Java的GUI系统，用来和Swing竞争，而开源集成开发环境Eclipse就是用Java和SWT开发的，具体见http://en.wikipedia.org/wiki/Standard_Widget_Toolkit。（本书脚注若无特殊说明均为译者注。）

② GTK+使用C语言开发，是类Unix系统下开发GUI应用程序的主流开发工具之一。它是自由软件，并是GNU计划的一部分，具体见http://zh.wikipedia.org/wiki/GTK。

③ REPL就是Read-Eval-Print Loop的缩写，意思是"阅读-评估-打印-循环"，它既可以作为独立的程序运行，也很容易作为程序的一部分使用。它为运行JavaScript脚本和查看结果提供了一种交互方式，详细内容见http://nodejs.org/docs/v0.6.6/api/repl.html 和http://nodejs.org/docs/v0.6.6/api/tty.html。

Web诞生之初，编写Web应用的工具还不够成熟。有些人尝试用Perl或TCL编写CGI脚本，PHP和Java语言刚刚被开发出来，而JavaScript甚至也被用于服务器端。Netscape的LiveWire服务端作为早期的Web应用服务器，就是用JavaScript编写的。微软ASP的某些版本使用的是JScript——它们自己的JavaScript实现。最近的一个服务器端JavaScript项目是在Java领域中使用的RingoJS应用框架。RingoJS构建于Rhino[①]之上，Rhino是一个用Java编写的JavaScript实现。

Node带来了一个前所未见的组合，那就是高速事件驱动I/O和V8高速JavaScript引擎；V8这个超快的JavaScript引擎是Google Chrome浏览器的核心。

1.2 为什么要使用 Node

JavaScript由于无处不在的浏览器而非常流行。它实现了许多现代高级语言的概念，比其他任何语言都不逊色。多亏了它的普及，软件行业才有大量经验丰富的JavaScript人才储备。

JavaScript是一门动态编程语言，拥有松散类型且可动态扩展的对象（能按需非正式地声明）。函数是一级对象，通常作为匿名闭包使用。这使得JavaScript比其他常用于编写Web应用的语言更加强大。理论上，这些特性使开发者的工作更加高效。平心而论，动态和非动态、静态类型和松散类型语言之间的优劣尚无定论，而且可能永远不会有定论。

JavaScript的一个短板是全局对象。所有的顶级变量都被扔给一个全局对象，这在混用多个模块时会导致难以预料的混乱。由于Web应用通常有大量的对象，且很可能是多个组织编写的，所以你自然会认为Node编程中的全局对象冲突会是个“雷区”。但其实不然，Node使用CommonJS[②]模块系统，这意味着模块的局部变量即使看起来像全局变量，实际上也是局部变量。这种模块间的清晰分离避免了全局对象的问题。

在Web应用服务器端和客户端使用同样的编程语言是人们由来已久的梦想。这个梦想可以追溯到早期的Java时代，那时Applet是用Java编写的服务器应用的前端，而对JavaScript的最初设想是将其作为Applet的一种轻量级脚本语言。但世事难料，到头来JavaScript取代Java成为在浏览器中使用的唯一语言。有了Node，在客户端和服务器端使用相同编程语言的梦想终于有望实现了，这门语言就是JavaScript。

语言在前后端通用有如下几个优势：

- 网线两端可能是相同的程序员；
- 代码能更容易地在服务器端和客户端间迁移；

① Rhino是用Java编写的JavaScript引擎，其设计目标是借助于强大的Java平台API简化JavaScript程序的编写。Rhino是Mozilla开发的自由软件，其最新的1.7r3版本实现了部分ECMAScript 5，可以从http://www.mozilla.org/rhino/下载它的源代码。

② CommonJS是一种规范，Node实现了这个规范。JavaScript是一门强大的、面向对象的语言，它有很多快速高效的解释器。JavaScript标准定义的API是为了构建基于浏览器的应用程序，不存在用于更广泛应用程序的标准库。CommonJS定义了构建很多普通应用程序（主要指非浏览器应用）的API，从而填补了这个空白。CommonJS的终极目标是提供一个类Python、Ruby和Java的标准库。这样的话，开发者可以使用CommonJS API编写应用程序，然后这些应用可以运行在不同的JavaScript解释器和不同的主机环境中。更多内容见http://www.commonjs.org/。