

<<Node.js开发指南>>

图书基本信息

书名：<<Node.js开发指南>>

13位ISBN编号：9787115283993

10位ISBN编号：7115283990

出版时间：2012-7

出版单位：人民邮电出版社

作者：郭家竇 (BYVoid)

页数：178

字数：249000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<Node.js开发指南>>

前言

这本书讲了什么 本书是一本 Node.js 的入门教程，写给想了解 Node.js 的开发人员。我的目标是使读者通过阅读本书，学会使用 Node.js 进行 Web 后端开发，同时能熟悉事件驱动的异步式编程风格，以便进一步了解 Node.js 的许多高级特性，以及它所应用的更多领域。

本书共6章，分别讨论了 Node.js 的背景、安装和配置方法、基本特性、核心模块以及一些进阶话题。

除此之外，还有2个附录，分别介绍了JavaScript的高级特性和 Node.js 编程规范。

下面简要概述各章的主要内容。

第1章“Node.js 简介”这一章概述了什么是 Node.js。

读过这章后，你将对 Node.js 有一个基本的认识，同时了解它与 JavaScript 的深厚渊源。

第2章“安装和配置 Node.js”这一章讲述了如何在各种不同的环境下安装和配置 Node.js 及其基本运行环境，同时你可以了解到如何编译 Node.js，以及多版本管理工具。

第3章“Node.js 快速入门”这一章讲解 Node.js 的基础知识，你将会学到如何使用 Node.js 的基本环境和工具进行开发、运行和调试。

同时，还会讲解异步式 I/O 与事件式编程的一些重要概念，这些概念将会贯穿全书。

此外这一章还详细介绍了 Node.js 的模块和包的系统，这些都是开发中经常会碰到的内容。

第4章“Node.js 核心模块”这一章以全局对象、基本工具、事件发射器、文件系统和 HTTP 为代表，介绍了 Node.js 最常用的核心模块。

你将会在后面的章节及以后的开发中经常与这些模块打交道。

第5章“使用 Node.js 进行 Web 开发”这一章是本书的实践性章节，一步一步教你如何从零开始用 Express 框架创建一个网站，实现路由控制、模板解析、会话管理、数据库访问等功能，最终创建一个 Web 2.0 微博网站。

第6章“Node.js 进阶话题”这一章涉及几个进阶话题，包括模块加载机制、控制流分析和优化、生产环境的应用部署等内容，最后还讨论了 Node.js 适用的范围，帮助读者在今后的开发中作出更好的取舍。

附录A“JavaScript 的高级特性”这个附录介绍了 JavaScript 的一些高级特性，如函数作用域、闭包和对象的操作等内容。

这些特性在浏览器端的 JavaScript 开发中并没有受到应有的重视，而在 Node.js 中却十分常见，阅读这个附录可以帮助你更好地理解并运用 JavaScript 进行复杂的网站开发。

附录B“Node.js 编程规范”这个附录介绍了 Node.js 代码风格的一些约定，遵守这些约定可以让你的代码更清晰、易懂，同时也有利于接口开发的统一。

该附录还分享了一些开发经验，可以让程序避免很多意外错误和性能损失。

谁应该阅读本书 本书的目标读者是想要学习 Node.js，但没有任何系统的经验的开发者。

如果你听说过 Node.js，并被它许多神奇的特性吸引，那么这本书就是为你准备的。

通过阅读本书，你可以对 Node.js 有全面的认识，学会如何用 Node.js 编程，了解事件驱动、异步式 I/O 的编程模式，同时还可以掌握一些使用 JavaScript 进行函数式编程的方法。

本书假设读者已经学过至少一门编程语言，对基本的程序设计语言概念（如变量、函数、递归、对象）有所了解。

如果你是首次学习编程语言，我建议你先学一门常见的且容易入门的语言，如 Java 或 C。

如何阅读本书 熟悉浏览器端 JavaScript 的读者将很容易学会 Node.js 的许多特性，包括事件式编程、闭包、回调函数等，因为这些特性已经在浏览器中被广泛应用。

同时，你还可以学到 Node.js 在 Web 开发中的服务器端与浏览器端的结合方式，这无论是对前端设计还是后端开发都是有利的。

你还会对 JavaScript 有一个全新的认识，因为服务端的 JavaScript 中没有 DOM 和 BOM，也不存在浏览器之间的兼容性问题。

不熟悉 JavaScript 但是了解 C、Java、C++、C# 的读者将很容易学会 JavaScript 的语言特性及 Node.js

<<Node.js开发指南>>

的基本机制，如模块和包。

你需要关注的仅仅是 JavaScript 语言的特别之处，以及服务器端开发中需要注意的一些要点。

已经非常了解 Web 后端开发（如 PHP、ASP.net、Ruby on Rails、Django 等）的读者，本书将通过 Node.js 给你一个不同的视野。

你会发现 Node.js 和这些传统的框架有很大的区别，因为它使用了事件式编程和异步 I/O，所以你需要改变一些已有的思维方式。

同时，你还能享受到 Web 前后端紧密配合带来的新鲜感，并可能对 Ajax 有全新的认识。

如果是完全没有接触过 JavaScript 的读者，那么我建议你看完本书的前两章以后，花点时间到 <http://www.w3school.com.cn/js/> 网站看看 JavaScript 的入门教程。

你只要了解基础知识就行了，本书并不要求你学成一个 JavaScript 专家。

在这之后请阅读本书的附录 A，了解一下实际开发中可能会遇到的稍微复杂的语言特性。

附录 A 是为本书量身定做的，你可以从中很快地学会 Node.js 经常使用到的那些特性。

如果你想更加深入系统地学习 JavaScript，推荐阅读 Mozilla JavaScript 指南 <http://developer.mozilla.org/en/JavaScript/Guide>。

本书从第 3 章开始，将介绍如何用 Node.js 开发，你应该仔细阅读这一章。

第 4 章是一些最基本的模块介绍，涉及 Node.js 模块的基本风格，这可能会帮助你理解后面介绍的 API

。第 5 章是一个真枪实弹的实战演练，跟随这一章的每个步骤你就可以用 Node.js 实现一个真正的 Web 应用，体验开发的成就感。

第 6 章则是一些进阶话题，你会在这里接触到 Node.js 的一些深层次概念，同时你还将学会如何真正部署 Node.js 应用。

本书的每一章最后都有一个参考资料小节，里面有很多有价值的资料，如果感兴趣不妨继续深入阅读。

在阅读本书的过程中，我建议你抽时间看看附录 B，在这里你会了解到 Node.js 开发的一些编程规范，写出符合社区风格的漂亮程序。

如何学习 Node.js 通读本书，你将会学到 Node.js 的很多东西，但如果想完全掌握它，我建议你自己亲自尝试运行本书中的每一段代码。

本书的所有代码可以在 <http://www.byvoid.com/project/node> 上找到。

除此之外，你最好自己用 Node.js 做一个项目，因为通过实践你会遇到很多问题，解决这些问题可以大大加深对 Node.js 的理解。

注意，不要忘了互联网网上的资源，比如 Node.js 的官方 API 文档 <http://nodejs.org/api/>。

我强烈推荐你去 CNodeJS 社区看看 <http://cnodejs.org/>，这里汇集了许许多多中国优秀的 Node.js 开发者。

他们每天都在讨论着大量有关 Node.js 各个方面的话题，你可以在上面获得很多帮助。

同时，CNodeJS 社区的网站也是用 Node.js 写成的，而且是开源的，它是一个非常好的让你了解如何用 Node.js 开发网站的实例。

体例说明 本书正文中出现的代码引用都会以等宽字体标出，例如：`console.log('Node.js')`。

代码段会以段落的形式用等宽字体显示，例如：

```
function hello() { console.log('Hello, world!'); }
```

 在正文之中，偶尔还会穿插一些提示和警告，例如： 这是一个提示。

这是一个警告。

致谢 感谢对这本书提出宝贵意见的朋友们，他们是牟瞳、李垚、周越、钟音、萧骐、杨旭东、孙嘉龙、范泽一、宋文杰、续本达、田劲锋、孟亚兰和李宇亮。

他们为本书的结构、内容、语言表述等方面给出了许多有建设性的意见。

感谢 CNodeJS 社区的贾超、田永强和微软亚洲研究院的杨懋，以及 VMware 公司的柴可夫。

他们不仅帮助审阅了本书，还解决了许多技术问题，给这本书提出了许多改进方案。

感谢弓辰开发的 Rime 输入法，我用它完成了本书的创作。

还要感谢我的朋友徐可可，图灵公司的杨海玲、谢工、王军花以及各位编辑，她们给我提供了许

多帮助和鼓舞，没有她们的激励，我很难顶着巨大的学业压力坚持写完这本书。

<<Node.js开发指南>>

内容概要

《Node.js开发指南》首先简要介绍Node.js，然后通过各种示例讲解Node.js的基本特性，再用案例式教学的方式讲述如何用Node.js进行Web开发，接着探讨一些Node.js进阶话题，最后展示如何将一个Node.js应用部署到生产环境中。

《Node.js开发指南》面向对Node.js感兴趣，但没有基础的读者，也可供已了解Node.js，并对Web前端/后端开发有一定经验，同时想尝试新技术的开发者参考。

<<Node.js开发指南>>

作者简介

郭家竇 (BYVoid)，清华大学计算机系2010级本科生，曾获得信息学奥林匹克竞赛 (NOI) 金牌。他从中学开始涉足开源开发，参与过“汉典网”等许多Web项目的前后端设计开发，同时是Linux输入法ibus-pinyin的作者。

他从2009年Node.js诞生之始，就一直在关注它的发展，有许多使用Node.js建立网站的经验，活跃于CNode社区。

<<Node.js开发指南>>

书籍目录

目 录

- 第1章 Node.js简介 1
 - 1.1 Node.js是什么 2
 - 1.2 Node.js能做什么 3
 - 1.3 异步式I/O与事件驱动 4
 - 1.4 Node.js的性能 5
 - 1.4.1 Node.js架构简介 5
 - 1.4.2 Node.js与PHP+Nginx 6
 - 1.5 JavaScript简史 6
 - 1.5.1 Netscape与LiveScript 7
 - 1.5.2 Java与Javascript 7
 - 1.5.3 微软的加入——JScript 8
 - 1.5.4 标准化——ECMAScript 8
 - 1.5.5 浏览器兼容性问题 9
 - 1.5.6 引擎效率革命和JavaScript的未来 9
 - 1.6 CommonJS 10
 - 1.6.1 服务端JavaScript的重生 10
 - 1.6.2 CommonJS规范与实现 11
 - 1.7 参考资料 12
- 第2章 安装和配置Node.js 13
 - 2.1 安装前的准备 14
 - 2.2 快速安装 14
 - 2.2.1 Microsoft Windows系统上安装Node.js 14
 - 2.2.2 Linux发行版上安装Node.js 16
 - 2.2.3 Mac OS X上安装Node.js 16
 - 2.3 编译源代码 17
 - 2.3.1 在POSIX系统中编译 17
 - 2.3.2 在Windows系统中编译 18
 - 2.4 安装Node包管理器 18
 - 2.5 安装多版本管理器 19
 - 2.6 参考资料 21
- 第3章 Node.js快速入门 23
 - 3.1 开始用Node.js编程 24
 - 3.1.1 Hello World 24
 - 3.1.2 Node.js命令行工具 25
 - 3.1.3 建立HTTP服务器 26
 - 3.2 异步式I/O与事件式编程 29
 - 3.2.1 阻塞与线程 29
 - 3.2.2 回调函数 31
 - 3.2.3 事件 33
 - 3.3 模块和包 34
 - 3.3.1 什么是模块 35
 - 3.3.2 创建及加载模块 35
 - 3.3.3 创建包 38
 - 3.3.4 Node.js包管理器 41

<<Node.js开发指南>>

- 3.4 调试 45
 - 3.4.1 命令行调试 45
 - 3.4.2 远程调试 47
 - 3.4.3 使用Eclipse调试Node.js 48
 - 3.4.4 使用node-inspector调试Node.js 54
- 3.5 参考资料 55
- 第4章 Node.js核心模块 57
 - 4.1 全局对象 58
 - 4.1.1 全局对象与全局变量 58
 - 4.1.2 process 58
 - 4.1.3 console 60
 - 4.2 常用工具util 61
 - 4.2.1 util.inherits 61
 - 4.2.2 util.inspect 62
 - 4.3 事件驱动events 63
 - 4.3.1 事件发射器 64
 - 4.3.2 error事件 65
 - 4.3.3 继承EventEmitter 65
 - 4.4 文件系统fs 65
 - 4.4.1 fs.readFile 66
 - 4.4.2 fs.readFileSync 67
 - 4.4.3 fs.open 67
 - 4.4.4 fs.read 68
 - 4.5 HTTP服务器与客户端 70
 - 4.5.1 HTTP服务器 70
 - 4.5.2 HTTP客户端 74
 - 4.6 参考资料 77
- 第5章 使用Node.js进行Web开发 79
 - 5.1 准备工作 80
 - 5.1.1 使用http模块 82
 - 5.1.2 Express框架 83
 - 5.2 快速开始 84
 - 5.2.1 安装Express 84
 - 5.2.2 建立工程 85
 - 5.2.3 启动服务器 86
 - 5.2.4 工程的结构 87
 - 5.3 路由控制 89
 - 5.3.1 工作原理 89
 - 5.3.2 创建路由规则 92
 - 5.3.3 路径匹配 93
 - 5.3.4 REST风格的路由规则 94
 - 5.3.5 控制权转移 95
 - 5.4 模板引擎 97
 - 5.4.1 什么是模板引擎 97
 - 5.4.2 使用模板引擎 98
 - 5.4.3 页面布局 99
 - 5.4.4 片段视图 100

<<Node.js开发指南>>

- 5.4.5 视图助手 100
- 5.5 建立微博网站 102
 - 5.5.1 功能分析 102
 - 5.5.2 路由规划 102
 - 5.5.3 界面设计 103
 - 5.5.4 使用Bootstrap 104
- 5.6 用户注册和登录 107
 - 5.6.1 访问数据库 107
 - 5.6.2 会话支持 110
 - 5.6.3 注册和登入 111
 - 5.6.4 页面权限控制 120
- 5.7 发表微博 123
 - 5.7.1 微博模型 123
 - 5.7.2 发表微博 125
 - 5.7.3 用户页面 126
 - 5.7.4 首页 127
 - 5.7.5 下一步 129
- 5.8 参考资料 129
- 第6章 Node.js进阶话题 131
 - 6.1 模块加载机制 132
 - 6.1.1 模块的类型 132
 - 6.1.2 按路径加载模块 132
 - 6.1.3 通过查找node_modules目录加载模块 133
 - 6.1.4 加载缓存 134
 - 6.1.5 加载顺序 134
 - 6.2 控制流 135
 - 6.2.1 循环的陷阱 135
 - 6.2.2 解决控制流难题 137
 - 6.3 Node.js应用部署 138
 - 6.3.1 日志功能 138
 - 6.3.2 使用cluster模块 140
 - 6.3.3 启动脚本 142
 - 6.3.4 共享80端口 143
 - 6.4 Node.js不是银弹 144
 - 6.5 参考资料 146
- 附录A JavaScript的高级特性 147
- 附录B Node.js编程规范 167
- 索引 175

<<Node.js开发指南>>

章节摘录

Node.js, 或者 Node, 是一个可以让 JavaScript 运行在服务器端的平台。它可以让 JavaScript 脱离浏览器的束缚运行在一般的服务器环境下, 就像运行 Python、Perl、PHP、Ruby 程序一样。

你可以用 Node.js 轻松地进行服务器端应用开发, Python、Perl、PHP、Ruby 能做的事情 Node.js 几乎都能做, 而且可以做得更好。

Node.js 是一个为实时 Web (Real-time Web) 应用开发而诞生的平台, 它从诞生之初就充分考虑了在实时响应、超大规模数据要求下架构的可扩展性。

这使得它摒弃了传统平台依靠多线程来实现高并发的设计思路, 而采用了单线程、异步式 I/O、事件驱动式的程序设计模型。

这些特性不仅带来了巨大的性能提升, 还减少了多线程程序设计的复杂性, 进而提高了开发效率。

Node.js 最初是由 Ryan Dahl 发起的开源项目, 后来被 Joyent 公司注意到。

Joyent 公司将 Ryan Dahl 招入旗下, 因此现在的 Node.js 由 Joyent 公司管理并维护。

尽管它诞生的时间 (2009年) 还不长, 但它的周围已经形成了一个庞大的生态系统。

Node.js 有着强大而灵活的包管理器 (node package manager, npm), 目前已经有上万个第三方模块, 其中有网站开发框架, 有 MySQL、PostgreSQL、MongoDB 数据库接口, 有模板语言解析、CSS 生成工具、邮件、加密、图形、调试支持, 甚至还有图形用户界面和操作系统 API 工具。

由 VMware 公司建立的云计算平台 Cloud Foundry 率先支持了 Node.js。

2011年6月, 微软宣布与 Joyent 公司合作, 将 Node.js 移植到 Windows, 同时 Windows Azure 云计算平台也支持 Node.js。

Node.js 目前还处在迅速发展阶段, 相信在不久的将来它一定会成为流行的 Web 应用开发平台。

让我们从现在开始, 一同探索 Node.js 的美妙世界吧!

1.1 Node.js 是什么 Node.js 不是一种独立的语言, 与 PHP、Python、Perl、Ruby 的“既是语言也是平台”不同。

Node.js 也不是一个 JavaScript 框架, 不同于 CakePHP、Django、Rails。

Node.js 更不是浏览器端的库, 不能与 jQuery、ExtJS 相提并论。

Node.js 是一个让 JavaScript 运行在服务端的开发平台, 它让 JavaScript 成为脚本语言世界的一等公民, 在服务端堪与 PHP、Python、Perl、Ruby 平起平坐。

Node.js 是一个划时代的技术, 它在原有的 Web 前端和后端技术的基础上总结并提炼出了许多新的概念和方法, 堪称是十多年来 Web 开发经验的集大成者。

Node.js 可以作为服务器向用户提供服务, 与 PHP、Python、Ruby on Rails 相比, 它跳过了 Apache、Nginx 等 HTTP 服务器, 直接面向前端开发。

Node.js 的许多设计理念与经典架构 (如 LAMP) 有着很大的不同, 可提供强大的伸缩能力, 以适应 21 世纪 10 年代以后规模越来越庞大的互联网环境。

Node.js 与 JavaScript 说起 JavaScript, 不得不让人想到浏览器。

传统意义上, JavaScript 是由 ECMAScript、文档对象模型 (DOM) 和浏览器对象模型 (BOM) 组成的, 而 Mozilla 则指出 JavaScript 由 Core JavaScript 和 Client JavaScript 组成。

之所以会有这种分歧, 是因为 JavaScript 和浏览器之间复杂的历史渊源, 以及其命途多舛的发展历程所共同造成的, 我们会在后面详述。

我们可以认为, Node.js 中所谓的 JavaScript 只是 Core JavaScript, 或者说是 ECMAScript 的一个实现, 不包含 DOM、BOM 或者 Client JavaScript。

这是因为 Node.js 不运行在浏览器中, 所以不需要使用浏览器中的许多特性。

Node.js 是一个让 JavaScript 运行在浏览器之外的平台。

它实现了诸如文件系统、模块、包、操作系统 API、网络通信等 Core JavaScript 没有或者不完善的功能。

历史上将 JavaScript 移植到浏览器外的计划不止一个, 但 Node.js 是最出色的一个。

<<Node.js开发指南>>

随着 Node.js 的成功，各种浏览器外的 JavaScript 实现逐步兴起，因此产生了 CommonJS 规范。CommonJS 试图拟定一套完整的 JavaScript 规范，以弥补普通应用程序所需的 API，譬如文件系统访问、命令行、模块管理、函数库集成等功能。

CommonJS 制定者希望众多服务端 JavaScript 实现遵循 CommonJS 规范，以便相互兼容和代码复用。Node.js 的部份实现遵循了 CommonJS 规范，但由于两者还都处于诞生之初的快速变化期，也会有不一致的地方。

Node.js 的 JavaScript 引擎是 V8，来自 Google Chrome 项目。

V8 号称是目前世界上最快的 JavaScript 引擎，经历了数次引擎革命，它的 JIT (Just-in-time Compilation，即时编译) 执行速度已经快到了接近本地代码的执行速度。

Node.js 不运行在浏览器中，所以也就不存在 JavaScript 的浏览器兼容性问题，你可以放心地使用 JavaScript 语言的所有特性。

……

<<Node.js开发指南>>

媒体关注与评论

“简洁的代码示例，轻快的语言，这本书带你进入同样简明的Node.js世界。如果你想立即使用Node.js进行Web开发，这里提供了绝佳的指导。”

——杨懋，微软亚洲研究院主管研究员 “本书是一本浅显易懂的Node.js入门读物，适合有一定JavaScript基础的开发人员阅读。读过这本书，你就完成了从学习Node.js相关知识，到使用Node.js构建实际Web系统的全过程。难能可贵的是，本书在讲解Node.js的同时，还详细介绍了Web开发领域的通用知识与原理，这些对开发完善健壮的Web应用必不可少。”

——贾超，淘宝网数据产品部资深经理，CNode社区发起者 “在CNode社区企盼将近两年后，第一本中文Node.js图书终于诞生了。跟着家骛同学的这本《Node.js开发指南》，你就会走进Node，初探到Node的好和美。”

——田永强（朴灵），淘宝网数据产品部门前端工程师，CNode社区组织者之一

<<Node.js开发指南>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>