

## <<Node.js入门经典>>

### 图书基本信息

书名：<<Node.js入门经典>>

13位ISBN编号：9787115311078

10位ISBN编号：7115311072

出版时间：2013-4

出版单位：人民邮电出版社

作者：奥尔波

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入门经典>>

### 内容概要

Node.js是一套用来编写高性能网络服务器的JavaScript工具包，从2009年诞生之日起，就获得了业内专家和技术社区的强烈关注。

而《Node.js入门经典》采用直观、循序渐进的方法对如何使用Node.js来开发及具速度和可扩展性优势的服务器端应用程序进行了讲解。

《Node.js入门经典》分为6部分，第1部分介绍了Node.js的基本概念和特性；第2部分讲解如何借助HTTP模块和Express Web框架，使用Node.js创建基本的网站；第3部分介绍了调试和测试Node.js应用程序的工具，以及部署Node.js应用的方法；第4部分讲解了Node.js实现实时编程的能力以及Socket.IO；第5部分介绍了Node.js API以及构建Node.js应用程序所使用的组件；第6部分则介绍了CoffeeScript这款JavaScript预编译器的知识，以及如何在Node.js中使用中间件、Backbone.js来创建单页面应用的知识。

。

## <<Node.js入门经典>>

### 作者简介

George Ornbo是英国的一位JavaScript和Ruby开发人员。他开发Web应用程序已有8年时间，一开始是以自由职业者的身份工作，最近则为伦敦的pebble {code}工作。他的博客地址是<http://shapedshed.com>，在网络中大多数常见的地方，他都以@shapedshed出现。

## 书籍目录

第1部分 入门 第1章 Node.js介绍 2 1.1 什么是Node.js 2 1.2 使用Node.js能做什么 3 1.3 安装并创建第一个Node.js程序 3 1.3.1 验证Node.js正确安装 4 1.3.2 创建“HelloWorld”Node.js程序 4 1.4 小结 5 1.5 问与答 6 1.6 测验 6 1.6.1 问题 6 1.6.2 答案 7 1.7 练习 7 第2章 npm (Node包管理器) 8 2.1 npm是什么 8 2.2 安装npm 9 2.3 安装模块 9 2.4 使用模块 10 2.5 如何找模块 11 2.5.1 官方来源 11 2.5.2 非官方来源 12 2.6 本地和全局的安装 13 2.6.1 本地安装 13 2.6.2 全局安装 13 2.7 如何找模块文档 14 2.8 使用package.json指定依赖关系 (dependency) 14 2.9 小结 16 2.10 问与答 16 2.11 测验 16 2.11.1 问题 16 2.11.2 答案 17 2.12 练习 17 第3章 Node.js的作用 18 3.1 设计Node.js的目的 18 3.2 理解I/O 19 3.3 处理输入 19 3.4 联网的I/O是不可预测的 22 3.5 人类是不可预测的 24 3.6 处理不可预测性 25 3.7 小结 26 3.8 问与答 26 3.9 测验 27 3.9.1 问题 27 3.9.2 答案 27 3.10 练习 27 第4章 回调 (Callback) 29 4.1 什么是回调 29 4.2 剖析回调 33 4.3 Node.js如何使用回调 34 4.4 同步和异步代码 36 4.5 事件循环 39 4.6 小结 39 4.7 问与答 39 4.8 测验 40 4.8.1 问题 40 4.8.2 答案 40 4.9 练习 40 第2部分 使用Node.js的基本网站 第5章 HTTP 44 5.1 什么是HTTP 44 5.2 使用Node.js的HTTP服务器 44 5.2.1 一个基础的服务器 44 5.2.2 加入头 (Header) 45 5.2.3 检查响应头 46 5.2.4 Node.js中的重定向 49 5.2.5 响应不同的请求 50 5.3 使用Node.js的HTTP客户端 52 5.4 小结 53 5.5 问与答 53 5.6 测验 54 5.6.1 问题 54 5.6.2 答案 54 5.7 练习 54 第6章 Express介绍 55 6.1 什么是Express 55 6.2 为什么使用Express 55 6.3 安装Express 56 6.4 创建一个基础的Express站点 56 6.5 探索Express 58 6.5.1 app.js 58 6.5.2 node\_modules 58 6.5.3 package.json 58 6.5.4 public 58 6.5.5 routes 58 6.5.6 views 58 6.6 介绍Jade 59 6.6.1 使用Jade定义页面结构 60 6.6.2 使用Jade输出数据 62 6.7 小结 68 6.8 问与答 68 6.9 测验 68 6.9.1 问题 69 6.9.2 答案 69 6.10 练习 69 第7章 深入Express 70 7.1 Web应用程序中的路由 70 7.2 在Express中路由如何工作 70 7.3 添加GET路由 71 7.4 添加POST路由 72 7.5 在路由中使用参数 73 7.6 让路由保持可维护性 74 7.7 视图渲染 75 7.8 使用本地变量 76 7.9 小结 78 7.10 问与答 78 7.11 测验 78 7.11.1 问题 79 7.11.2 答案 79 7.12 练习 79 第8章 数据的持久化 80 8.1 什么是持久的数据 80 8.2 将数据写入文件 81 8.3 从文件读取数据 82 8.4 读取环境变量 83 8.5 使用数据库 84 8.5.1 关系数据库 84 8.5.2 NoSQL数据库 85 8.6 在Node.js中使用MongoDB 85 8.6.1 安装MongoDB 86 8.6.2 连接MongoDB 87 8.6.3 定义文档 89 8.6.4 将TwitterBootstrap包含进来 90 8.6.5 索引 (Index) 视图 91 8.6.6 创建 (Create) 视图 93 8.6.7 编辑视图 95 8.6.8 删除任务 98 8.6.9 添加闪出消息 99 8.6.10 验证输入的数据 101 8.7 小结 102 8.8 问与答 103 8.9 测验 103 8.9.1 问题 103 8.9.2 答案 103 8.10 练习 104 第3部分 调试、测试与部署 第9章 调试Node.js应用程序 106 9.1 调试 106 9.2 STUDIO模块 107 9.3 Node.js调试器 111 9.4 NodeInspector 113 9.5 关于测试的注释 116 9.6 小结 116 9.7 问与答 116 9.8 测验 117 9.8.1 问题 117 9.8.2 答案 117 9.9 练习 117 第10章 测试Node.js应用程序 119 10.1 为什么测试 119 10.2 Assert (断言) 模块 120 10.3 第三方测试工具 122 10.4 行为驱动的开发 (BehaviorDrivenDevelopment) 125 10.4.1 Vows 125 10.4.2 Mocha 128 10.5 小结 131 10.6 问与答 131 10.7 测验 132 10.7.1 问题 132 10.7.2 答案 132 10.8 练习 132 第11章 部署Node.js应用程序 133 11.1 准备好部署 133 11.2 在云上托管 133 11.3 Heroku 135 11.3.1 注册Heroku 135 11.3.2 为Heroku准备应用程序 136 11.3.3 将应用程序部署到Heroku 137 11.4 CloudFoundry 138 11.4.1 注册CloudFoundry 138 11.4.2 为CloudFoundry准备应用程序 139 11.4.3 将应用程序部署到CloudFoundry 140 11.5 Nodester 141 11.5.1 注册Nodester 141 11.5.2 为Nodester准备应用程序 142 11.5.3 将应用程序部署到Nodester 143 11.6 其他PaaS提供商 144 11.7 小结 144 11.8 问与答 144 11.9 测验 145 11.9.1 测验 145 11.9.2 答案 145 11.10 练习 145 第4部分 使用Node.js的中间站点 第12章 介绍Socket.IO 148 12.1 现在要开始学习一些完全不同的技术了 148 12.2 动态Web简史 148 12.3 Socket.IO 149 12.4 基础的Socket.IO示例 150 12.5 从服务器发送数据到客户端 152 12.6 将数据广播给客户端 156 12.7 双向数据 160 12.8 小结 163 12.9 问与答 163 12.10 测验 164 12.10.1 问题 164 12.10.2 答案 164 12.11 练习 165 第13章 一个Socket.IO聊天服务器 166 13.1 Express和Socket.IO 166 13.2 添加昵称 168 13.2.1 将昵称发送给服务器 169 13.2.2 管理昵称列表 171 13.2.3 使用回调来验证 174 13.2.4 广播昵称列表 178 13.2.5 添加消息收发功能 179 13.3 小结 183 13.4 问与答 184 13.5 测验 184 13.5.1 问题 184 13.5.2 答案 184 13.6 练习 185 第14章 一个流Twitter客户端 186 14.1 流API 186 14.2 注册Twitter 187 14.3 和Node.js一起使用Twitter的API 189 14.4 从数据中挖掘含义 191 14.5 将数据推送到浏览器 194 14.6 创建一个实时的爱恨表 197 14.7 小结 206 14.8 问与答 206 14.9 测验 206 14.9.1 问题 206 14.9.2 答案 206 14.10 练习 207 第15章

JSONAPI 208 15.1 API 208 15.2 JSON 209 15.3 使用Node.js发送JSON数据 211 15.4 从JavaScript对象创建JSON 212 15.5 使用Node.js消费JSON数据 213 15.6 使用Node.js创建JSONAPI 216 15.6.1 在Express中以JSON发送数据 216 15.6.2 构建应用程序 219 15.7 小结 224 15.8 问与答 225 15.9 测验 225 15.9.1 问题 225 15.9.2 答案 225 15.10 练习 226 第5部分 探索Node.jsAPI 第16章 进程模块 228 16.1 进程是什么 228 16.2 退出进程以及进程中的错误 230 16.3 进程与信号 230 16.4 向进程发送信号 231 16.5 使用Node.js创建脚本 233 16.6 给脚本传递参数 234 16.7 小结 236 16.8 问与答 236 16.9 测验 237 16.9.1 问题 237 16.9.2 答案 237 16.10 练习 238 第17章 子进程模块 239 17.1 什么是子进程 239 17.2 杀死子进程 241 17.3 与子进程通信 242 17.4 集群 ( Cluster ) 模块 244 17.5 小结 246 17.6 问与答 246 17.7 测验 246 17.7.1 问题 246 17.7.2 答案 246 17.8 练习 247 第18章 事件模块 248 18.1 理解事件 248 18.2 通过HTTP演示事件 251 18.3 用事件玩乒乓 254 18.4 动态编写事件侦听器程序 255 18.5 小结 258 18.6 问与答 258 18.7 测验 259 18.7.1 问题 259 18.7.2 答案 259 18.8 练习 259 第19章 缓冲区模块 260 19.1 二进制数据初步 260 19.2 从二进制到文本 261 19.3 二进制和Node.js 262 19.4 Node.js中的缓冲区是什么 ? 264 19.5 写入缓冲区 265 19.6 向缓冲区追加数据 266 19.7 复制缓冲区 267 19.8 修改缓冲区中的字符串 267 19.9 小结 268 19.10 问与答 268 19.11 测验 268 19.11.1 问题 268 19.11.2 答案 269 19.12 练习 269 第20章 流模块 270 20.1 流简介 270 20.2 可读流 272 20.3 可写流 275 20.4 通过管道连接流 276 20.5 流的MP3 277 20.6 小结 278 20.7 问与答 278 20.8 测验 279 20.8.1 问题 279 20.8.2 答案 279 20.9 练习 279 第6部分 进一步的Node.js 开发 第21章 CoffeeScript 282 21.1 什么是CoffeeScript 282 21.2 安装与运行CoffeeScript 284 21.3 为什么要使用预编译器 285 21.4 CoffeeScript的功能 286 21.4.1 最小语法 286 21.4.2 条件和比较 287 21.4.3 循环 288 21.4.4 字符串 289 21.4.5 对象 290 21.4.6 类、继承和super 291 21.5 调试CoffeeScript 294 21.6 对CoffeeScript的反应 294 21.7 小结 295 21.8 问与答 295 21.9 测验 296 21.9.1 问题 296 21.9.2 答案 296 21.10 练习 296 第22章 创建Node.js模块 298 22.1 为什么创建模块 298 22.2 流行的Node.js模块 298 22.3 package.json文件 299 22.4 文件夹结构 301 22.5 开发和测试模块 302 22.6 添加可执行文件 304 22.7 使用面向对象或者基于原型的编程 305 22.8 通过GitHub共享代码 306 22.9 使用TravisCI 307 22.10 发布到npm 309 22.11 公开模块 310 22.12 小结 310 22.13 问与答 310 22.14 测验 311 22.14.1 问题 311 22.14.2 答案 311 22.15 练习 311 第23章 使用Connect创建中间件 312 23.1 什么是中间件 312 23.2 Connect中的中间件 313 23.3 使用中间件的访问控制 317 23.4 按IP地址限制访问 319 23.5 将用户强制到单个域上 321 23.6 小结 323 23.7 问与答 324 23.8 测验 324 23.8.1 问题 324 23.8.2 答案 324 23.9 练习 324 23.9.1 问题 324 23.9.2 答案 324 23.9.3 练习 324 23.9.4 问题 324 23.9.5 答案 324 第24章 结合使用Backbone.js与Node.js 326 24.1 什么是Backbone.js 326 24.2 Backbone.js如何工作 327 24.3 一个简单的Backbone.js视图 332 24.4 使用Backbone.js创建记录 336 24.5 小结 337 24.6 问与答 337 24.7 测验 338 24.7.1 问题 338 24.7.2 答案 338 24.8 练习 338

## 章节摘录

版权页：插图：对于那些从Python和Ruby这样的语言切换过来的人而言，CoffeeScript大体上是受欢迎的。

语法相似，无需花费许多年来学习JavaScript所有的怪癖。

CoffeeScript用一套熟悉的语法做了所有的重活，输出的是一个经过策划的JavaScript版本，避免落入许多常见的编程陷阱。

对于那些编程经验不多的人而言，CoffeeScript可能不流行。

它是令人发指的软件工具清单中的另外一个需要学习的东西。

尤其对于那些通过jQuery对JavaScript有些经验的人，编译到JavaScript的额外步骤会是慢吞吞并且不必要的。

对于有经验的JavaScript程序员而言，什么反应都有。

这些程序员花费了许多年的时间来理解错综复杂的JavaScript并花费数小时时间来完成痛苦的调试工作！

虽然CoffeeScript会避免许多常见的JavaScript问题，但它输出的是带有自己的看法的JavaScript，可能并不与有经验的JavaScript开发人员的风格一致。

此外，有些人将CoffeeScript看作是不相干的编译，如果他们已经是技术高超的JavaScript开发人员的话。

它经常被看作是用来帮助经验不足的开发人员的玩具。

除了调试更复杂以外，大家还认为用CoffeeScript编写Node.js项目的话，和其他开发人员共事会难得多。

使用CoffeeScript，就假定项目中的其他所有人都理解CoffeeScript并且预编译JavaScript。

大家的指责是这样的：CoffeeScript给想要在项目上协同工作的人添加了另一道障碍。

JavaScript的创建者Brendan Eich在他的网站上写了如下文字（<http://brendaneich.com/2011/01/harmony-of-my-dreams/>）：CoffeeScript做得不错，要比JS更方便，只要你对Python式的有意义的空格和从另外一个源语言生成JS带来的开销买账。

这是一个生动的引述。

CoffeeScript要比JavaScript更方便，但其代价就是要学习一种新语法并且需要额外的编译步骤。

21.7小结 在本章，我们了解了CoffeeScript这一JavaScript的预编译器。

我们了解到可以在Node.js项目中使用它，并且检查了诸如语法、循环、对象和类这样的功能。

我们了解了使用CoffeeScript与编写原生JavaScript之间的优缺点。

我们了解了CoffeeScript所能提供的生产力提升、调试和协作上的问题以及CoffeeScript如何绕开JavaScript编程语言中更为困难的那些部分。

21.8问与答 问：别当骑墙派了！

我到底该不该用CoffeeScript？

答：我还是骑墙吧！

简而言之就是要看情况。

要看你的JavaScript技艺水平如何，是否与其他知道CoffeeScript的开发人员一起协作，以及是否喜欢CoffeeScript。

CoffeeScript当然可以带来许多方便，但它也不是没有潜在的问题。

## <<Node.js入门经典>>

### 编辑推荐

《Node.js入门经典》内容循序渐进、深入浅出、步骤详尽，而且附有大量适合动手实践的示例，可帮助读者在最短的时间内掌握Node.js。

本书适合对Node.js感兴趣的零基础人员阅读，也适合对Web前端开发、后端开发感兴趣的技术人员阅读。

读者可通过<http://vdisk.weibo.com/s/s7fmW>下载《Node.js入门经典》的所有源代码。

## <<Node.js入门经典>>

### 名人推荐

这是一本绝佳的Node.js图书，她涵盖了Node.js平台的所有方面，并通过案例式教学来帮助读者理解、掌握Node.js的使用。

——Andrew Nesbitt, Forward Internet Group开发人员



## <<Node.js入门经典>>

### 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>