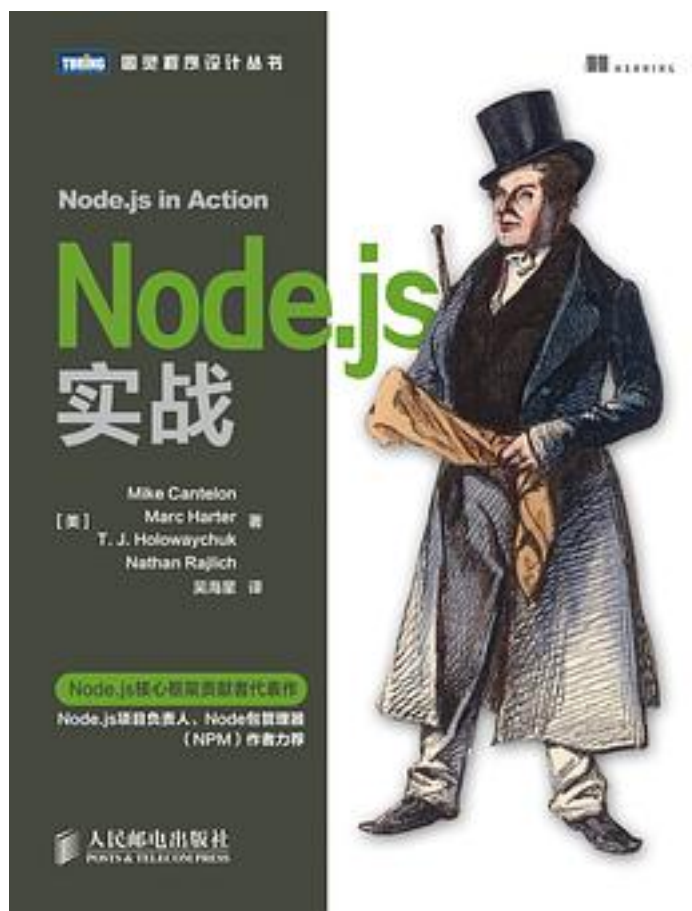


Node.js实战



[Node.js实战 下载链接1](#)

著者:[美] Mike Cantelon

出版者:人民邮电出版社

出版时间:2014-5

装帧:平装

isbn:9787115352460

服务器端JavaScript? 没错。Node.js是一个JavaScript服务器，支持可伸缩的高性能Web应用。借助异步I/O，这个服务器可以同时做很多事情，能满足聊天、游戏和实时统计等应用的需求。并且既然是JavaScript，那你就可以全栈使用一种语言。

本书向读者展示了如何构建产品级应用，对关键概念的介绍清晰明了，贴近实际的例子，涵盖从安装到部署的各个环节，是一部讲解与实践并重的优秀著作。通过学习本书，读者将深入异步编程、数据存储、输出模板、读写文件系统，掌握创建TCP/IP服务器和命令行工具等非HTTP程序的技术。本书同样非常适合熟悉Rails、Django或PHP开发的读者阅读学习。

本书主要内容：

Node.js及其扩展的安装配置；

全面理解异步编程和事件循环；

学会开发微博、聊天和游戏等热门应用。

作者介绍:

作者简介：

Mike Cantelon

Node.js核心框架贡献者、Node社区活跃分子、资深培训师和演讲人。

Marc Harter

Node.js核心框架贡献者。

T.J. Holowaychuk

参与开发了很多Node.js模块，包括流行的Express框架。

Nathan Rajlich

大名鼎鼎的TooTallNate，Node.js核心代码提交者。

译者简介：

吴海星

2001年毕业于南京理工大学。编程数载代码不过几十万，翻译几年码字不过几百万。项目不过十几个，带队不到五十人。年过而立，惴惴不安，愈加发奋，孜孜求学，愿凭绵薄之力，贡献于IT社区。

目录: 第一部分 Node基础

第1章 欢迎进入Node.js世界 2

1.1 构建于JavaScript之上 3

1.2 异步和事件触发：浏览器 4

1.3 异步和事件触发：服务器 5

1.4 DIRT程序 6

1.5 默认DIRT 8

1.5.1 简单的异步程序 9

1.5.2 Hello World HTTP服务器 10

1.5.3 流数据 10

1.6 小结	11
第2章 构建有多个房间的聊天室程序	12
2.1 程序概览	12
2.2 程序需求及初始设置	15
2.2.1 提供HTTP和WebSocket服务	15
2.2.2 创建程序的文件结构	16
2.2.3 指明依赖项	16
2.2.4 安装依赖项	17
2.3 提供HTML、CSS和客户端JavaScript的服务	18
2.3.1 创建静态文件服务器	19
2.3.2 添加HTML和CSS文件	21
2.4 用Socket.IO处理与聊天相关的消息	23
2.4.1 设置Socket.IO服务器	24
2.4.2 处理程序场景及事件	25
2.5 在程序的用户界面上使用客户端JavaScript	29
2.5.1 将消息和昵称/房间变更请求传给服务器	29
2.5.2 在用户界面中显示消息及可用房间	30
2.6 小结	34
第3章 Node编程基础	35
3.1 Node功能的组织及重用	36
3.1.1 创建模块	37
3.1.2 用module.exports微调模块的创建	39
3.1.3 用node_modules重用模块	40
3.1.4 注意事项	41
3.2 异步编程技术	42
3.2.1 用回调处理一次性事件	43
3.2.2 用事件发射器处理重复性事件	46
3.2.3 异步开发的难题	53
3.3 异步逻辑的顺序化	54
3.3.1 什么时候使用串行流程控制	55
3.3.2 实现串行化流程控制	56
3.3.3 实现并行化流程控制	58
3.3.4 利用社区里的工具	60
3.4 小结	61
第二部分 用Node开发Web程序	
第4章 构建Node Web程序	64
4.1 HTTP服务器的基础知识	64
4.1.1 Node如何向开发者呈现HTTP请求	65
4.1.2 一个用“Hello World”做响应的HTTP服务器	67
4.1.3 读取请求头及设定响应头	67
4.1.4 设定HTTP响应的状态码	67
4.2 构建RESTful Web服务	68
4.2.1 用POST请求创建资源	69
4.2.2 用GET请求获取资源	71
4.2.3 用DELETE请求移除资源	72
4.3 提供静态文件服务	73
4.3.1 创建一个静态文件服务器	73
4.3.2 处理服务器错误	77
4.3.3 用fs.stat()实现先发制人的错误处理	77
4.4 从表单中接受用户输入	78
4.4.1 处理提交的表单域	78
4.4.2 用formidable处理上传的文件	81
4.4.3 计算上传进度	84
4.5 用HTTPS加强程序的安全性	85

- 4.6 小结 86
- 第5章 存储Node程序中的数据 87
 - 5.1 无服务器的数据存储 88
 - 5.1.1 内存存储 88
 - 5.1.2 基于文件的存储 88
 - 5.2 关系型数据库管理系统 91
 - 5.2.1 MySQL 91
 - 5.2.2 PostgreSQL 99
 - 5.3 NoSQL数据库 100
 - 5.3.1 Redis 101
 - 5.3.2 MongoDB 105
 - 5.3.3 Mongoose 107
 - 5.4 小结 109
- 第6章 Connect 110
 - 6.1 搭建一个Connect程序 111
 - 6.2 Connect的工作机制 112
 - 6.2.1 做日志的中间件 112
 - 6.2.2 响应“hello world”的中间件 113
 - 6.3 为什么中间件的顺序很重要 114
 - 6.3.1 中间件什么时候不调用next() 114
 - 6.3.2 用中间件的顺序执行认证 115
 - 6.4 挂载中间件和服务 115
 - 6.4.1 认证中间件 116
 - 6.4.2 显示管理面板的中间件 117
 - 6.5 创建可配置中间件 118
 - 6.5.1 创建可配置的logger中间件组件 119
 - 6.5.2 构建路由中间件组件 120
 - 6.5.3 构建一个重写URL的中间件组件 122
 - 6.6 使用错误处理中间件 123
 - 6.6.1 Connect的默认错误处理器 124
 - 6.6.2 自行处理程序错误 124
 - 6.6.3 使用多个错误处理中间件组件 125
 - 6.7 小结 129
- 第7章 Connect自带的中间件 130
 - 7.1 解析cookie、请求主体和查询字符串的中间件 131
 - 7.1.1 cookieParser(): 解析HTTP cookie 131
 - 7.1.2 bodyParser(): 解析请求主体 134
 - 7.1.3 limit(): 请求主体的限制 135
 - 7.1.4 query(): 查询字符串解析 137
 - 7.2 实现Web程序核心功能的中间件 138
 - 7.2.1 logger(): 记录请求 138
 - 7.2.2 favicon(): 提供favicon 140
 - 7.2.3 methodOverride(): 伪造HTTP方法 141
 - 7.2.4 vhost(): 虚拟主机 143
 - 7.2.5 session(): 会话管理 144
 - 7.3 处理Web程序安全的中间件 148
 - 7.3.1 basicAuth(): HTTP基本认证 148
 - 7.3.2 csrf(): 跨站请求伪造防护 150
 - 7.3.3 errorHandler(): 开发错误处理 150
 - 7.4 提供静态文件服务的中间件 152
 - 7.4.1 static(): 静态文件服务 152
 - 7.4.2 compress(): 压缩静态文件 154
 - 7.4.3 directory(): 目录列表 156
 - 7.5 小结 157

- 第8章 Express 158
 - 8.1 生成程序骨架 160
 - 8.1.1 安装Express的可执行程序 161
 - 8.1.2 生成程序 162
 - 8.1.3 探索程序 162
 - 8.2 配置Express和你的程序 164
 - 8.3 渲染视图 166
 - 8.3.1 视图系统配置 167
 - 8.3.2 视图查找 169
 - 8.3.3 把数据输出到视图中 171
 - 8.4 处理表单和文件上传 175
 - 8.4.1 实现照片模型 175
 - 8.4.2 创建照片上传表单 176
 - 8.4.3 显示上传照片列表 178
 - 8.5 创建资源下载 179
 - 8.5.1 创建照片下载路由 179
 - 8.5.2 实现照片下载路由 180
 - 8.6 小结 182
- 第9章 Express进阶 183
 - 9.1 认证用户 184
 - 9.1.1 保存和加载用户 184
 - 9.1.2 注册新用户 189
 - 9.1.3 已注册用户登录 194
 - 9.1.4 用户加载中间件 197
 - 9.2 先进的路由技术 199
 - 9.2.1 校验用户内容提交 199
 - 9.2.2 特定路由中间件 202
 - 9.2.3 实现分页 205
 - 9.3 创建一个公开的REST API 208
 - 9.3.1 设计API 208
 - 9.3.2 添加基本的认证 209
 - 9.3.3 实现路由 209
 - 9.3.4 启用内容协商 212
 - 9.4 错误处理 214
 - 9.4.1 处理404错误 215
 - 9.4.2 处理错误 217
 - 9.5 小结 220
- 第10章 测试Node程序 221
 - 10.1 单元测试 222
 - 10.1.1 assert模块 222
 - 10.1.2 Nodeunit 225
 - 10.1.3 Mocha 227
 - 10.1.4 Vows 232
 - 10.1.5 should.js 234
 - 10.2 验收测试 235
 - 10.2.1 Tobi 236
 - 10.2.2 Soda 237
 - 10.3 小结 239
- 第11章 Web程序模板 240
 - 11.1 用模板保持代码的整洁性 240
 - 11.2 嵌入JavaScript的模板 244
 - 11.2.1 创建模板 245
 - 11.2.2 用EJS过滤器处理模板数据 246
 - 11.2.3 将EJS集成到你的程序中 249

- 11.2.4 在客户端程序中使用EJS 250
- 11.3 使用Mustache模板语言与Hogan 251
 - 11.3.1 创建模板 251
 - 11.3.2 Mustache标签 252
 - 11.3.3 微调Hogan 254
- 11.4 用Jade做模板 255
 - 11.4.1 Jade基础知识 256
 - 11.4.2 Jade模板中的逻辑 258
 - 11.4.3 组织Jade模板 260
- 11.5 小结 264
- 第三部分 在Node中更进一步
- 第12章 部署Node程序并维持正常运行时间 266
 - 12.1 安置Node程序 266
 - 12.1.1 专用的和虚拟私有服务器 267
 - 12.1.2 云主机 268
 - 12.2 部署的基础知识 269
 - 12.2.1 从Git存储库部署 270
 - 12.2.2 让Node保持运行 270
 - 12.3 让正常运行时间和性能达到最优 271
 - 12.3.1 用Upstart维护正常运行时间 272
 - 12.3.2 集群API：利用多核的优势 273
 - 12.3.3 静态文件及代理 275
 - 12.4 小结 277
- 第13章 超越Web服务器 278
 - 13.1 Socket.IO 278
 - 13.1.1 创建一个最小的Socket.IO程序 279
 - 13.1.2 用Socket.IO触发页面和CSS的重新加载 281
 - 13.1.3 Socket.IO的其他用法 283
 - 13.2 深入TCP/IP网络 284
 - 13.2.1 处理缓冲区和二进制数据 284
 - 13.2.2 创建TCP服务器 286
 - 13.2.3 创建TCP客户端 289
 - 13.3 跟操作系统交互的工具 290
 - 13.3.1 单例的全局process对象 291
 - 13.3.2 使用文件系统模块 293
 - 13.3.3 繁衍外部进程 296
 - 13.4 开发命令行工具 301
 - 13.4.1 解析命令行参数 301
 - 13.4.2 处理stdin和stdout 302
 - 13.4.3 添加彩色的输出 304
 - 13.5 小结 306
- 第14章 Node生态系统 307
 - 14.1 给Node开发人员的在线资源 308
 - 14.1.1 Node和模块的参考资料 308
 - 14.1.2 Google群组 309
 - 14.1.3 IRC 309
 - 14.1.4 GitHub问题列表 310
 - 14.2 GitHub 310
 - 14.2.1 GitHub入门 311
 - 14.2.2 添加一个项目到GitHub中 312
 - 14.2.3 用GitHub协作 314
 - 14.3 为npm库做贡献 316
 - 14.3.1 准备包 317
 - 14.3.2 编写包规范 317

14.3.3 测试和发布包 318
14.4 小结 320
附录A 安装Node和社区附加组件 321
附录B 调试Node 329
附录C Express的扩展及配置 336
• • • • • ([收起](#))

[Node.js实战_下载链接1_](#)

标签

Node.js

JavaScript

nodejs

web开发

node.js

前端开发

编程

node

评论

很赞。集众大牛于一书。实战好帮手~

挺好的入门书

unfinish

讲解的知识点很多，比一般的node.JS书容量要大，这也导致了有些知识点讲解不是那么细，理解起来比较费力，有些地方的组织顺序有点混乱，不是循序渐进的引导，不过确实是本好书，可以学到不少别的书中不会提到的东东

挺不错的一本书

相当不错的一本书，整个Node的生态介绍的非常完整，大致让你了解与学会整个Node技术栈，作为Node入门不可多得的好书。

node和异步编程的入门好书！

node.js 入门推荐。看完后继续看 《MEAN Web Development》

很棒

比较全面的入门书

清晰明白

期待Go in Action

还行吧，看了之后基本对nodejs就算入门了。后三分之一的内容有点杂。

很不错的Node入门书

有些翻译问题。和其他in action系列差不多。适合做入门nodejs的第二本书。

部分框架类代码有点过时（比如SOCKET.IO、EXPRESS这些），但是思想不过时，KOA还是主要基于中间件的思想。不知道能不能等到第二版把代码更新一下。

挺多实际的代码，书有点旧，有些代码因为库更新现在已经运行不出来了，有问题还得上网搜，express高级那章没看懂，还是水平比较渣。。

清库存计划 3.30读完

实在是深入浅出的感觉。虽然前面几章节把node.js的主要原理讲完就差不多了，但后面关于connect, express的还不错。

快速翻完了一遍。node很棒，作为前端开发，node给了我写api的机会，实现自己的兴趣又更近了一步，同时不会花费太多的迁移成本。不过专业的事还是交给专业的软件，node不适合静态服务，除非你只是想了解原理。推荐学习node官方文档，koa2，翻一遍这本书，基本上可以满足需求了。

[Node.js实战_下载链接1](#)

书评

这本书还没出版的时候，我已经通读过了，是试读版的。对于你所说的这些东西，我理解为都是末枝末节。这本书的重点不是教你语法，学JS语法可以看《权威指南》，学nodeAPI可以看官方文档，学express也可以去看文档，网上例子还很多。这本书最大的价值就在于它介绍了NODE WEB APPL...

内容比较多，放到了blog里：

<http://vernonzheng.com/2014/12/28/NodeJS%E5%AE%9E%E6%88%98%E7%BB%8F%E9%AA%8C%E6%80%BB%E7%BB%93/> 笔记目录： 1. 1、require 2. 2、exports与module.exports的区别 3. 3、模块缓存与猴子补丁： 4. 4、Node两种常用的响应逻辑组织方式 5. 5、减...

我之所以了解到这本书是在 expressjs.com 的主页上看到了推荐，其次我发现作者是 TJ，于是我猜想这一定是一本对于了解 Node.js 的不二教材了。一开始我看书里的有些代码别扭时，我也没觉得什么，心想作者可能对于一些细枝末节没有弄得那么仔细，后来越看才越发现，这书就是一...

[Node.js实战_下载链接1](#)