

<<深入浅出Node.js>>

图书基本信息

书名：<<深入浅出Node.js>>

13位ISBN编号：9787115335500

10位ISBN编号：7115335508

出版时间：2013-12

出版时间：人民邮电出版社

作者：朴灵

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>

<<深入浅出Node.js>>

内容概要

本书从不同的视角介绍了 Node 内在的特点和结构。由首章Node 介绍为索引，涉及Node 的各个方面，主要内容包含模块机制的揭示、异步I/O 实现原理的展现、异步编程的探讨、内存控制的介绍、二进制数据Buffer 的细节、Node 中的网络编程基础、Node 中的Web 开发、进程间的消息传递、Node 测试以及通过Node 构建产品需要的注意事项。最后的附录介绍了Node 的安装、调试、编码规范和NPM 仓库等事宜。

本书适合想深入了解 Node 的人员阅读。

<<深入浅出Node.js>>

作者简介

朴灵
真名田永强，文艺型码农，就职于阿里巴巴数据平台，资深工程师，Node.js布道者，写了多篇文章介绍Node.js的细节。
活跃于CNode社区，是线下会议NodeParty的组织者和JSConf China（沪JS和京JS）的组织者之一。
热爱开源，多个Node.js模块的作者。
个人GitHub地址：<http://github.com/JacksonTian>。
叩首问路，码梦为生。

<<深入浅出Node.js>>

书籍目录

第1章 Node简介	1
1.1 Node的诞生历程	1
1.2 Node的命名与起源	1
1.2.1 为什么是JavaScript	2
1.2.2 为什么叫Node	2
1.3 Node给JavaScript带来的意义	2
1.4 Node的特点	4
1.4.1 异步I/O	4
1.4.2 事件与回调函数	6
1.4.3 单线程	7
1.4.4 跨平台	7
1.5 Node的应用场景	8
1.5.1 I/O密集型	8
1.5.2 是否不擅长CPU密集型业务	8
1.5.3 与遗留系统和平共处	10
1.5.4 分布式应用	10
1.6 Node的使用者	10
1.7 参考资源	11
第2章 模块机制	12
2.1 CommonJS规范	13
2.1.1 CommonJS的出发点	13
2.1.2 CommonJS的模块规范	14
2.2 Node的模块实现	15
2.2.1 优先从缓存加载	16
2.2.2 路径分析和文件定位	16
2.2.3 模块编译	18
2.3 核心模块	20
2.3.1 JavaScript核心模块的编译过程	21
2.3.2 C/C++核心模块的编译过程	22
2.3.3 核心模块的引入流程	25
2.3.4 编写核心模块	25
2.4 C/C++扩展模块	27
2.4.1 前提条件	28
2.4.2 C/C++扩展模块的编写	29
2.4.3 C/C++扩展模块的编译	30
2.4.4 C/C++扩展模块的加载	31
2.5 模块调用栈	32
2.6 包与NPM	33
2.6.1 包结构	34
2.6.2 包描述文件与NPM	34
2.6.3 NPM常用功能	37
2.6.4 局域NPM	42
2.6.5 NPM潜在问题	43
2.7 前后端共用模块	44
2.7.1 模块的侧重点	44

<<深入浅出Node.js>>

2.7.2	AMD规范	44
2.7.3	CMD规范	45
2.7.4	兼容多种模块规范	45
2.8	总结	46
2.9	参考资源	46
第3章	异步I/O	47
3.1	为什么要异步I/O	47
3.1.1	用户体验	48
3.1.2	资源分配	49
3.2	异步I/O实现现状	50
3.2.1	异步I/O与非阻塞I/O	50
3.2.2	理想的非阻塞异步I/O	54
3.2.3	现实的异步I/O	54
3.3	Node的异步I/O	56
3.3.1	事件循环	56
3.3.2	观察者	56
3.3.3	请求对象	57
3.3.4	执行回调	59
3.3.5	小结	60
3.4	非I/O的异步API	60
3.4.1	定时器	60
3.4.2	process.nextTick()	61
3.4.3	setImmediate()	62
3.5	事件驱动与高性能服务器	63
3.6	总结	65
3.7	参考资源	65
第4章	异步编程	66
4.1	函数式编程	66
4.1.1	高阶函数	66
4.1.2	偏函数用法	67
4.2	异步编程的优势与难点	68
4.2.1	优势	69
4.2.2	难点	70
4.3	异步编程解决方案	74
4.3.1	事件发布/订阅模式	74
4.3.2	Promise/Deferred模式	82
4.3.3	流程控制库	93
4.4	异步并发控制	105
4.4.1	bagpipe的解决方案	105
4.4.2	async的解决方案	109
4.5	总结	110
4.6	参考资源	110
第5章	内存控制	111
5.1	V8的垃圾回收机制与内存限制	111
5.1.1	Node与V8	112
5.1.2	V8的内存限制	112
5.1.3	V8的对象分配	112

<<深入浅出Node.js>>

5.1.4	V8的垃圾回收机制	113
5.1.5	查看垃圾回收日志	119
5.2	高效使用内存	121
5.2.1	作用域	121
5.2.2	闭包	123
5.2.3	小结	124
5.3	内存指标	124
5.3.1	查看内存使用情况	124
5.3.2	堆外内存	126
5.3.3	小结	127
5.4	内存泄漏	127
5.4.1	慎将内存当做缓存	127
5.4.2	关注队列状态	130
5.5	内存泄漏排查	130
5.5.1	node-heapdump	131
5.5.2	node-memwatch	132
5.5.3	小结	135
5.6	大内存应用	135
5.7	总结	136
5.8	参考资源	136
第6章	理解Buffer	137
6.1	Buffer结构	137
6.1.1	模块结构	137
6.1.2	Buffer对象	138
6.1.3	Buffer内存分配	139
6.2	Buffer的转换	141
6.2.1	字符串转Buffer	141
6.2.2	Buffer转字符串	142
6.2.3	Buffer不支持的编码类型	142
6.3	Buffer的拼接	143
6.3.1	乱码是如何产生的	144
6.3.2	setEncoding()与string_decoder()	144
6.3.3	正确拼接Buffer	145
6.4	Buffer与性能	146
6.5	总结	149
6.6	参考资源	149
第7章	网络编程	150
7.1	构建TCP服务	150
7.1.1	TCP	150
7.1.2	创建TCP服务器端	151
7.1.3	TCP服务的事件	153
7.2	构建UDP服务	154
7.2.1	创建UDP套接字	154
7.2.2	创建UDP服务器端	154
7.2.3	创建UDP客户端	155
7.2.4	UDP套接字事件	155
7.3	构建HTTP服务	155

<<深入浅出Node.js>>

7.3.1	HTTP	156
7.3.2	http模块	157
7.3.3	HTTP客户端	161
7.4	构建WebSocket服务	163
7.4.1	WebSocket握手	164
7.4.2	WebSocket数据传输	167
7.4.3	小结	169
7.5	网络服务与安全	169
7.5.1	TLS/SSL	170
7.5.2	TLS服务	172
7.5.3	HTTPS服务	173
7.6	总结	175
7.7	参考资源	176
第8章	构建Web应用	177
8.1	基础功能	177
8.1.1	请求方法	178
8.1.2	路径解析	179
8.1.3	查询字符串	180
8.1.4	Cookie	181
8.1.5	Session	184
8.1.6	缓存	190
8.1.7	Basic认证	193
8.2	数据上传	195
8.2.1	表单数据	195
8.2.2	其他格式	196
8.2.3	附件上传	197
8.2.4	数据上传与安全	199
8.3	路由解析	201
8.3.1	文件路径型	202
8.3.2	MVC	202
8.3.3	RESTful	207
8.4	中间件	210
8.4.1	异常处理	214
8.4.2	中间件与性能	215
8.4.3	小结	216
8.5	页面渲染	217
8.5.1	内容响应	217
8.5.2	视图渲染	219
8.5.3	模板	220
8.5.4	Bigpipe	231
8.6	总结	235
8.7	参考资源	235
第9章	玩转进程	236
9.1	服务模型的变迁	236
9.1.1	石器时代：同步	236
9.1.2	青铜时代：复制进程	237
9.1.3	白银时代：多线程	237

<<深入浅出Node.js>>

9.1.4	黄金时代：事件驱动	237
9.2	多进程架构	238
9.2.1	创建子进程	239
9.2.2	进程间通信	240
9.2.3	句柄传递	242
9.2.4	小结	247
9.3	集群稳定之路	248
9.3.1	进程事件	248
9.3.2	自动重启	249
9.3.3	负载均衡	254
9.3.4	状态共享	255
9.4	Cluster模块	257
9.4.1	Cluster工作原理	258
9.4.2	Cluster事件	259
9.5	总结	259
9.6	参考资源	260
第10章	测试	261
10.1	单元测试	261
10.1.1	单元测试的意义	261
10.1.2	单元测试介绍	263
10.1.3	工程化与自动化	276
10.1.4	小结	277
10.2	性能测试	278
10.2.1	基准测试	278
10.2.2	压力测试	280
10.2.3	基准测试驱动开发	281
10.2.4	测试数据与业务数据的转换	283
10.3	总结	284
10.4	参考资源	284
第11章	产品化	285
11.1	项目工程化	285
11.1.1	目录结构	285
11.1.2	构建工具	286
11.1.3	编码规范	289
11.1.4	代码审查	289
11.2	部署流程	290
11.2.1	部署环境	291
11.2.2	部署操作	291
11.3	性能	293
11.3.1	动静分离	293
11.3.2	启用缓存	294
11.3.3	多进程架构	294
11.3.4	读写分离	295
11.4	日志	295
11.4.1	访问日志	295
11.4.2	异常日志	296
11.4.3	日志与数据库	299

<<深入浅出Node.js>>

11.4.4	分割日志	299
11.4.5	小结	299
11.5	监控报警	299
11.5.1	监控	300
11.5.2	报警的实现	302
11.5.3	监控系统的稳定性	303
11.6	稳定性	303
11.7	异构共存	304
11.8	总结	305
11.9	参考资源	305
附录A	安装Node	306
A.1	Windows系统下的Node安装	306
A.2	Mac系统下Node的安装	307
A.3	Linux系统下Node的安装	308
A.4	总结	309
A.5	参考资源	309
附录B	调试Node	310
B.1	Debugger	310
B.2	Node Inspector	311
B.2.1	安装Node Inspector	312
B.2.2	错误堆栈	312
B.3	总结	313
附录C	Node编码规范	314
C.1	根源	314
C.2	编码规范	315
C.2.1	空格与格式	315
C.2.2	命名规范	317
C.2.3	比较操作	318
C.2.4	字面量	318
C.2.5	作用域	318
C.2.6	数组与对象	319
C.2.7	异步	320
C.2.8	类与模块	320
C.2.9	注解规范	321
C.3	最佳实践	321
C.3.1	冲突的解决原则	321
C.3.2	给编辑器设置检测工具	321
C.3.3	版本控制中的hook	322
C.3.4	持续集成	322
C.4	总结	322
C.5	参考资源	323
附录D	搭建局域NPM仓库	324
D.1	NPM仓库的安装	325
D.1.1	安装Erlang和CouchDB	325
D.1.2	搭建NPM仓库	326
D.2	高阶应用	328
D.2.1	镜像仓库	328

<<深入浅出Node.js>>

D.2.2	私有模块应用	328
D.2.3	纯私有仓库	329
D.3	总结	331
D.4	参考资源	332

<<深入浅出Node.js>>

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>