

Broadview®
www.broadview.com.cn

OpenResty

完全开发指南

构建百万级别并发的Web应用

罗剑锋 著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

OpenResty

完全开发指南

构建百万级别并发的Web应用

罗剑锋 著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

OpenResty 是一个基于 Nginx 的高性能 Web 平台,能够方便地搭建处理超高并发的动态 Web 应用、Web 服务和动态网关。

与现有的其他开发语言/环境相比,OpenResty 有着高性能、高灵活性、易于学习和扩展等许多优点,近年来得到了越来越多开发者的关注,也有了很成功的应用范例,如 Adobe、Dropbox、GitHub 等知名公司都基于 OpenResty 构建了自己的后端业务应用。

OpenResty 自带完善的帮助文档,开发社区也很活跃,但相关的学习资料——特别是中文资料较少。本书基于作者多年使用 OpenResty 的经验,系统地阐述了 OpenResty 相关的各方面知识和要点,帮助读者快速掌握这个高效易用的 Web 开发平台,进而实现 HTTP/HTTPS/TCP/UDP 等多种网络应用。

本书结构严谨、详略得当,具有较强的实用性,适合广大软件开发工程师、系统运维工程师、编程爱好者和计算机专业学生阅读参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

OpenResty 完全开发指南:构建百万级别并发的 Web 应用 / 罗剑锋著. —北京:电子工业出版社,2018.9
ISBN 978-7-121-34896-9

I. ①O… II. ①罗… III. ①互联网络—网络服务器—程序设计 IV. ①TP368.5

中国版本图书馆 CIP 数据核字(2018)第 188129 号

策划编辑:孙学瑛

责任编辑:牛 勇

印 刷:北京天宇星印刷厂

装 订:北京天宇星印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×980 1/16 印张:14.25 字数:317 千字

版 次:2018 年 9 月第 1 版

印 次:2018 年 9 月第 1 次印刷

定 价:69.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式:010-51260888-819, faq@phei.com.cn。

前言

随感

本书肇始于三年多前我的《Nginx 模块开发指南》一书。最初是以书末的附录形式出现，只有短短的几页，粗略地介绍了 OpenResty 的核心组件 ngx_lua。连我自己也没有想到，几年后的今天它竟然“脱胎换骨”，进化成了一本颇具规模的正式图书。

写作本书还是有感于目前国内技术书刊市场的现状。

十几二十几年前只有少数资深专家掌握核心技术，通过著书立说的方式来分享知识，普惠大众。但随着互联网的高速发展，知识的获取方式变得越来越简单了，任何人都可以在网上轻松地查找到所需的资料，也可以在网上很容易地发表文章。书——曾经被誉为“进步的阶梯”“精神的食粮”——已经不是那么重要了。

另一方面，互联网的普及也降低了书的严肃性和出版门槛。个人“恶意推测”，也许是为了“图省事”或者“赚快钱”，有相当多的人只是把若干博客文章集合在一起，再加以少量修改就“攒”成了一本书。这种“乱炖”“杂烩”形式的书籍拼凑的痕迹十分明显，缺乏内在的逻辑和连贯性，不过凭借着网络上积累的“人气”也能够获得不错的销量，但在我看来实在是对读者的不尊重和不负责。

一个极端的例子是前段时间偶然遇到的名为《□□开发实战》的书，其粗制滥造程度简直是“超乎想象”，“不料，也不信竟会凶残到这地步”^①——几乎 90% 的内容都是原封不动地拷贝网络上现成的材料，再有就是直接复制数十页全无注释的杂乱代码，通篇看起来可能只有前言是“原创”，着实是“钦佩”该书作者厚颜无耻的“功力”。

^① 原文出自鲁迅《纪念刘和珍君》。

单纯地感慨“世风日下”“人心不古”是没有用的，我所能做的，就是尽自己“微茫”的努力，写出一些无愧于己于人的文字。

关于 OpenResty

有这样一种说法：“Nginx 是网络世界里的操作系统，而 OpenResty 则是 Nginx 上的 Web 服务器”。

Nginx 在 Web Server 业内的领军地位早已经得到了公认，是高性能服务器的杰出代表。它采用 C 语言开发，能够跨平台运行，把性能挖掘优化技术发挥到了极致。正因为如此，Nginx 也很自然地成为了一个超越原生操作系统的开发平台，程序员可以完全无视底层系统之间的差异，在 Nginx 的框架里调用丰富的数据结构和功能接口，开发出高性能高可移植的各种应用程序。^①

但基于 Nginx 开发主要使用的语言是 C/C++，开发难度高周期长，虽然没有达到“望而生畏”的程度但亦不远矣。好在 OpenResty 应运而生，在 Nginx 里嵌入了 LuaJIT 环境和 Lua 语言，就如同给裸系统添加了一个高效易用的 Shell，瞬间就让 Nginx 开发的难度直线下降，降低到了普通的心智模型可以理解掌握的水平。

早期 OpenResty 对于自身的定位主要还是 HTTP Server（其实也是受到 Nginx 的限制），可以利用“胶水语言”Lua 来操纵 Nginx，灵活定制业务逻辑，方便快捷地搭建出超高并发的各种 Web 服务，从而节约时间和人力成本。多年来的实践证明，这方面它的确工作得非常出色。

近两年 OpenResty 的发展开始加速，支持了 TCP/UDP 协议，扩充了众多的专用库、应用框架以及外围工具，逐渐形成了一个比较独立自洽的生态体系。虽然 Nginx 仍然是核心，但看得出 OpenResty 有淡化自身“Nginx Bundle”色彩的趋势，力图成为一个更伟大的存在。

随着软件基金会和商业公司的成立，OpenResty 获得了前所未有的成长动力。“路远，正未有穷期”，在此借本书送上诚挚的祝福与期待。

^① 通常来说 Nginx 适合运行单线程的 I/O 密集型应用，但实际上它也可以使用多线程技术运行 CPU 密集型应用。

致谢

首先要感谢 Nginx 的作者 Igor Sysoev 和 OpenResty 的作者 agentzh, 正是因为他们多年来持续无私的奉献, 我们才能够拥有如此强大易用的 Web Server。

接下来我要感谢父母多年来的养育之恩, 感谢妻子和两个可爱的女儿(“点心”组合)在生活中的陪伴, 愿你们能够永远幸福快乐。

我也要感谢读者选择本书, 希望读者能从中汲取有用的知识, 让 OpenResty 成为工作中的得力助手。

您的朋友 罗剑锋

2018 年 7 月 18 日 于 北京 798 园区

目录

第 0 章 导读.....	1	1.13 总结	22
0.1 关于本书	1	第 2 章 Nginx 平台.....	23
0.2 读者对象	1	2.1 简介	23
0.3 读者要求	3	2.2 进程模型.....	24
0.4 运行环境	3	2.3 配置文件.....	25
0.5 本书的结构	3	2.4 变量	26
0.6 如何阅读本书	5	2.5 HTTP 服务	27
0.7 本书的源码	5	2.5.1 server 配置	28
第 1 章 总论.....	7	2.5.2 location 配置	28
1.1 简介	7	2.6 TCP/UDP 服务	29
1.2 历史	8	2.7 反向代理.....	29
1.3 组成	9	2.7.1 上游集群	30
1.4 版本	11	2.7.2 代理转发	31
1.5 安装	12	2.8 运行日志.....	31
1.5.1 直接安装	12	2.8.1 访问日志	32
1.5.2 源码安装	13	2.8.2 错误日志	32
1.5.3 定制安装	13	2.9 总结	32
1.6 目录结构	14	第 3 章 Lua 语言.....	35
1.7 启停服务	15	3.1 简介	35
1.8 组件管理工具	15	3.2 注释	36
1.9 命令行工具	16	3.3 数据类型.....	36
1.10 参考手册	18	3.4 字符串	37
1.11 性能对比	18	3.5 变量	38
1.12 应用架构	21	3.6 运算	39

3.6.1 算术运算	39	3.13.2 保护调用	58
3.6.2 关系运算	39	3.13.3 可变参数	59
3.6.3 逻辑运算	40	3.14 总结	59
3.6.4 字符串运算	40	第4章 LuaJIT 环境	61
3.6.5 注意事项	41	4.1 简介	61
3.7 控制语句	41	4.2 goto 语句	62
3.7.1 语句块	41	4.3 jit 库	62
3.7.2 赋值语句	41	4.4 table 库	63
3.7.3 分支语句	42	4.5 bit 库	63
3.7.4 循环语句	43	4.6 ffi 库	65
3.8 函数	44	4.7 编译为字节码	67
3.8.1 定义函数	44	4.8 编译为机器码	68
3.8.2 参数和返回值	45	4.9 总结	68
3.9 表	46	第5章 开发概述	71
3.9.1 定义表	46	5.1 应用示例	71
3.9.2 操作表	46	5.1.1 编码实现	71
3.9.3 范围循环	47	5.1.2 测试验证	73
3.9.4 作为函数的参数	48	5.2 运行命令	74
3.10 模块	48	5.3 目录结构	75
3.11 面向对象	49	5.4 配置指令	76
3.11.1 基本特性	49	5.5 运行机制	77
3.11.2 原型模式	50	5.5.1 处理阶段	77
3.11.3 self 参数	51	5.5.2 执行程序	79
3.12 标准库	51	5.5.3 定时任务	81
3.12.1 base 库	52	5.5.4 流程图	81
3.12.2 package 库	52	5.6 功能接口	83
3.12.3 string 库	53	5.7 核心库	83
3.12.4 table 库	54	5.8 应用开发流程	84
3.12.5 math 库	55	5.9 总结	85
3.12.6 io 库	56	第6章 基础功能	87
3.12.7 os 库	57	6.1 系统信息	87
3.12.8 debug 库	57	6.2 运行日志	88
3.12.9 使用技巧	57	6.3 时间日期	89
3.13 高级特性	58		
3.13.1 闭包	58		

6.3.1	当前时间	90	7.6	请求行	111
6.3.2	时间戳	90	7.6.1	版本	112
6.3.3	格式化时间戳	90	7.6.2	方法	112
6.3.4	更新时间	91	7.6.3	地址	112
6.3.5	睡眠	91	7.6.4	参数	113
6.4	数据编码	92	7.7	请求头	114
6.4.1	Base64	92	7.7.1	读取数据	114
6.4.2	JSON	92	7.7.2	改写数据	115
6.4.3	MessagePack	94	7.8	请求体	115
6.5	正则表达式	95	7.8.1	丢弃数据	115
6.5.1	配置指令	95	7.8.2	读取数据	115
6.5.2	匹配选项	96	7.8.3	改写数据	116
6.5.3	匹配	96	7.9	响应头	117
6.5.4	查找	98	7.9.1	改写数据	117
6.5.5	替换	99	7.9.2	发送数据	118
6.5.6	切分	100	7.9.3	过滤数据	118
6.6	高速缓存	101	7.10	响应体	118
6.6.1	创建缓存	101	7.10.1	发送数据	118
6.6.2	使用缓存	102	7.10.2	过滤数据	119
6.7	总结	103	7.11	手动收发数据	120
第 7 章	HTTP 服务	105	7.12	流程控制	121
7.1	简介	105	7.12.1	重定向请求	121
7.2	配置指令	106	7.12.2	终止请求	121
7.3	常量	107	7.13	检测断连	122
7.3.1	状态码	107	7.14	综合示例	123
7.3.2	请求方法	108	7.15	总结	126
7.4	变量	108	第 8 章	访问后端	127
7.4.1	读变量	108	8.1	简介	127
7.4.2	写变量	109	8.2	子请求	128
7.5	基本信息	110	8.2.1	接口说明	128
7.5.1	请求来源	110	8.2.2	应用示例	129
7.5.2	起始时间	110	8.2.3	使用建议	130
7.5.3	请求头	110	8.3	协程套接字	131
7.5.4	暂存数据	111	8.3.1	配置指令	131

8.3.2	创建对象	132	8.8.5	复用连接	150
8.3.3	超时设置	133	8.8.6	简单查询	150
8.3.4	建立连接	133	8.8.7	高级查询	152
8.3.5	复用连接	134	8.8.8	防止 SQL 注入	152
8.3.6	关闭连接	134	8.9	总结	153
8.3.7	发送数据	135	第 9 章 反向代理	155	
8.3.8	接收数据	135	9.1	简介	155
8.3.9	应用示例	136	9.2	上游集群	156
8.4	DNS 客户端	137	9.2.1	静态服务器信息	157
8.4.1	创建对象	138	9.2.2	动态服务器信息	158
8.4.2	查询地址	138	9.2.3	服务器下线	159
8.4.3	缓存地址	139	9.2.4	当前上游集群	159
8.5	HTTP 客户端	140	9.3	负载均衡	160
8.5.1	创建对象	140	9.3.1	使用方式	160
8.5.2	发送请求	140	9.3.2	功能接口	161
8.6	WebSocket 客户端	142	9.4	总结	162
8.6.1	创建对象	142	第 10 章 高级功能	163	
8.6.2	建立连接	143	10.1	共享内存	163
8.6.3	关闭连接	143	10.1.1	配置指令	163
8.6.4	复用连接	143	10.1.2	写操作	164
8.6.5	发送数据	144	10.1.3	读操作	165
8.6.6	接收数据	144	10.1.4	删除操作	166
8.7	Redis 客户端	145	10.1.5	计数操作	166
8.7.1	创建对象	145	10.1.6	队列操作	166
8.7.2	建立连接	145	10.1.7	过期操作	167
8.7.3	关闭连接	146	10.1.8	其他操作	168
8.7.4	复用连接	146	10.2	定时器	168
8.7.5	执行命令	146	10.2.1	配置指令	168
8.7.6	管道	147	10.2.2	单次任务	169
8.7.7	脚本	148	10.2.3	周期任务	170
8.8	MySQL 客户端	148	10.3	进程管理	171
8.8.1	创建对象	149	10.3.1	进程类型	171
8.8.2	建立连接	149	10.3.2	工作进程	172
8.8.3	服务器版本号	150			
8.8.4	关闭连接	150			

10.3.3 监控进程	173	11.7.2 Session Tickets.....	193
10.3.4 特权进程	173	11.8 总结	193
10.4 轻量级线程	174	第 12 章 HTTP2 服务	195
10.4.1 启动线程	175	12.1 简介	195
10.4.2 等待线程	175	12.2 服务配置.....	196
10.4.3 挂起线程	176	12.3 应用开发.....	197
10.4.4 停止线程	177	12.4 测试验证.....	197
10.4.5 信号量	178	12.5 总结	198
10.5 总结	179	第 13 章 WebSocket 服务.....	199
第 11 章 HTTPS 服务	181	13.1 简介	199
11.1 简介	181	13.2 服务配置.....	200
11.1.1 密码学	181	13.3 应用开发.....	200
11.1.2 网络协议	182	13.4 总结	202
11.2 服务配置	184	第 14 章 TCP/UDP 服务	203
11.3 应用开发	185	14.1 简介	203
11.4 基本信息	185	14.2 配置指令.....	204
11.4.1 协议版本号	185	14.3 运行机制.....	205
11.4.2 主机名	186	14.3.1 处理阶段	205
11.4.3 地址	186	14.3.2 执行程序	206
11.5 加载证书	187	14.3.3 流程图	206
11.5.1 清除证书	187	14.4 功能接口.....	208
11.5.2 设置证书	187	14.5 应用示例.....	208
11.5.3 设置私钥	188	14.6 总结	210
11.5.4 测试验证	189	第 15 章 结束语	211
11.6 查验证书	189	附录 A 推荐书目	215
11.6.1 发送查询	189	附录 B 定制 OpenResty	217
11.6.2 通知客户端	191		
11.7 会话复用	191		
11.7.1 Session ID	191		

第 0 章

导读

0.1 关于本书

OpenResty 是一个基于 Nginx 的高性能 Web 平台，能够方便地使用动态脚本语言 Lua 搭建高并发、高扩展性的 Web 应用和动态网关。

与 Go、PHP、Python、Node.js 等现有的其他 Web 开发语言/环境相比，OpenResty 具有高性能、高灵活性、易于学习和扩展等许多优点，在高性能 Web 编程领域得到了广泛的应用，已经有为数众多的国内外大公司基于 OpenResty 构建了自己的业务服务，例如 Adobe、DropBox、GitHub 等。

虽然 OpenResty 自带完善的帮助文档，开发社区也很活跃，但它毕竟还是一个较新的开发平台，相关的学习资料较少，了解的人不多。有鉴于此，作者基于多年使用 OpenResty 的经验编写了本书，希望能够为 OpenResty 的普及尽一份自己的力量，也希望读者能够利用 OpenResty 开发出更多更好的 Web 应用。

0.2 读者对象

本书适合以下各类读者：

- 追求高性能的 Web 应用研发工程师；
- 微服务、API 网关、Web 应用防火墙的研发工程师；
- 手机游戏、网络游戏后端服务器的研发工程师；
- 通用的 HTTP/TCP/UDP 应用服务研发工程师；
- 工作在 Linux 系统上的运维、测试工程师；

■ 计算机编程爱好者和在校学生。

随着宽带网络的快速普及和移动互联网的高速发展，网站需要为越来越多的用户提供服务，处理越来越多的并发请求，要求服务器必须具有很高的性能才能应对不断增长的需求和突发的访问高峰。在超高并发请求的场景下，很多常用的服务开发框架都会显得“力不从心”，服务能力严重下降，很难优化。这时我们就可以选择 OpenResty，它内置高效的事件驱动模型和进程池机制，直接在“起点”上构建高性能的 Web 应用，而且可以很容易地调整配置参数进一步释放系统潜力。

现在的网站架构大都是分布式系统，经常部署有成百上千个内部模块，这些模块通常基于“微服务”“服务网格”等架构，彼此之间的联系十分复杂，在演化过程中系统会逐渐变得难以理解和维护。OpenResty 具有优秀的反向代理和负载均衡能力，在复杂的分布式系统中可以充当 API Gateway 的角色，分治整合不同种类的服务简化系统，并使用内嵌的 Lua 脚本添加缓冲、限流、防护、认证等额外功能。

游戏类应用是目前互联网特别是移动互联网上的一大类应用，用户量很庞大。这些应用多数会提供在线服务，要求后端稳定可靠，最好还能够快速修复错误或上线新功能。OpenResty 非常适合担当这样的后端服务器，它不仅性能高运行稳定，而且由于使用的是动态脚本语言 Lua，上手容易且功能丰富，能够很好地缩短产品的开发周期，实现快速迭代。此外，因为 Lua 语言也经常被用于游戏客户端开发，使用 OpenResty 还可以打通前后端，统一开发语言，轻松地成为掌控全局的“全栈”工程师。

除了开发标准的 Web 应用，OpenResty 也是一个通用的服务器开发框架，内部结构良好，基础设施完备，支持 HTTP/HTTPS/WebSocket/TCP/UDP 等多种网络协议和 Redis、MySQL 等常见的数据库，能够使用 Lua 以简单易理解的同步非阻塞模式编程，快捷实现各种业务逻辑，部分或完全取代自研框架或其他开源框架，实现任意的后台服务。

OpenResty 不仅是一个单纯的服务器软件，它还是一个完整的应用环境，其中就包含了一个非常有用的 Lua 脚本解释器。运维、测试工程师可以使用小巧灵活的 Lua 代替 Shell、Perl、Python 等语言，调用 OpenResty 的内部功能接口写出高效实用的各种脚本，如系统管理、状态监控、单元测试或压力测试，方便自己的工作。

最关键的一点，OpenResty 是完全开源的，拥有成熟活跃的开发社区和许多顶级的开发者，研究、参与它可以领会真正的“开源”精神，学习到各种前沿的理念、技术和知识，提高自身的能力。

0.3 读者要求

本书不要求读者具备专门的编程语言知识。

开发 Web 服务通常给人的感觉是很难，要掌握复杂的开发技巧和应用框架，但 OpenResty 把这个门槛大大降低了。它的开发模型非常直观容易理解，而且使用的主力编程语言是 Lua，一种小巧轻便的动态脚本语言，学习难度很低，只要具有初级编程经验就可以快速掌握。

但如果想要深入 OpenResty 底层编写出更高效的代码，还是建议读者多了解一些 C 语言和 Linux 的相关知识。

0.4 运行环境

OpenResty 可以跨平台编译和运行，支持 Linux、FreeBSD、macOS、Windows 等多种操作系统。但就目前市场来看，Linux 是应用最普及的服务器操作系统，故本书的开发环境选用 Linux。

Linux 有很多的发行版本，企业中使用较多的是偏重于稳定性的 CentOS 和偏重于易用性的 Ubuntu，出于个人喜好的原因本书选择了 Ubuntu，版本号是 16.04.03。

0.5 本书的结构

对于大多数读者来说，OpenResty 可能都是一个“陌生”的开发环境，所以本书采用循序渐进的方式组织全书的章节：首先介绍基本知识作为入门，然后解析运行机制和开发流程，再由浅入深地逐步讲解功能接口和如何开发各种 Web 服务。

全书共 15 章，各章的内容简介如下。

■ 第 1 章：总论

本章简要介绍 OpenResty 的历史、组成和编译安装的方法。

■ 第 2 章：Nginx 平台

Nginx 是 OpenResty 的核心部件，本章介绍了它的特点、进程模型和各种应用服务的配置方法。

■ 第 3 章：Lua 语言

本章讲解 OpenResty 的工作语言 Lua，包括详细的语法和标准库。

■ 第4章：LuaJIT 环境

本章介绍 OpenResty 使用的 Lua 运行环境 LuaJIT，它的运行效率更高，而且提供很多特别的优化和库，比原生的 Lua 更加强大。

■ 第5章：开发概述

本章在宏观的层次介绍开发 OpenResty 应用的基本流程、配置指令、运行机制等知识，帮助读者从总体上理解掌握 OpenResty。

■ 第6章：基础功能

本章介绍 OpenResty 里的一些基础功能，如系统信息、日志、时间日期、编码格式转换、正则表达式、高速缓存等。

■ 第7章：HTTP 服务

本章介绍 OpenResty 为开发 HTTP 服务提供的大量功能接口，操纵 HTTP 请求和响应，学习完本章就能够轻松开发出高性能的 Web 应用。

■ 第8章：访问后端

本章介绍 OpenResty 提供的两种高效通信机制：`location.capture` 和 `cosocket`，还有基于它们实现的一些客户端库，可以访问 HTTP、Redis、MySQL 等多种后端。

■ 第9章：反向代理

本章介绍 OpenResty 的反向代理功能，搭建动态网关，并使用 `ngx.upstream` 和 `ngx.balancer` 实现深度定制。

■ 第10章：高级功能

本章介绍 OpenResty 里的共享内存、定时器、进程管理和轻量级线程这四个高级功能。

■ 第11章：HTTPS 服务

本章介绍如何在 OpenResty 里开发 HTTPS 服务，实践动态加载证书、动态查验证书和会话复用等 HTTPS 优化技术。

■ 第12章：HTTP2 服务

本章介绍如何在 OpenResty 里开发 HTTP2 服务。

■ 第13章：WebSocket 服务

本章介绍如何在 OpenResty 里开发 WebSocket 服务。

■ 第 14 章：TCP/UDP 服务

本章介绍 OpenResty 里处理 TCP/UDP 协议的 stream 子系统，能够基于 TCP/UDP 协议开发出更通用的 Web 服务。

■ 第 15 章：结束语

本章给出了读者在阅读完本书后进一步学习研究 OpenResty 的方向。

0.6 如何阅读本书

初次接触 OpenResty 的读者应当先阅读第 1 章至第 4 章，在本书的指导下安装配置 OpenResty，搭建自己的开发环境，了解 OpenResty 的组成、运行平台和工作语言，然后再学习后续的章节。

如果读者已经比较熟悉 OpenResty，那么可以跳过前 4 章，从第 5 章开始顺序阅读，或者检索目录直接跳到感兴趣的章节，研究 OpenResty 的内部运行机制、功能接口、各种服务的配置和开发方式。

0.7 本书的源码

为方便读者利用本书学习研究 OpenResty，作者在 GitHub 网站上发布了本书内所有示例程序的源代码，地址是：

https://github.com/chronolaw/openresty_dev.git #OpenResty 开发示例

对于想深入钻研的读者，还有另外两个项目供进一步参考：

https://github.com/chronolaw/annotated_nginx.git #Nginx 源码详细注解
<https://github.com/chronolaw/favorite-nginx.git> #各种有用的 Nginx 资源

