

Go Web 编程 Go Web Programming

〔新加坡〕郑兆雄 (Sau Sheong Chang) 著
黄健宏 译

Go Web 编程

Go Web Programming

[新加坡] 郑兆雄 (Sau Sheong Chang) 著
黄健宏 译

人 民 邮 电 出 版 社
北 京

图书在版编目 (CIP) 数据

Go Web编程 / (新加坡) 郑兆雄著 ; 黄健宏译. --
北京 : 人民邮电出版社, 2017. 12
书名原文: Go Web Programming
ISBN 978-7-115-32247-0

I. ①G… II. ①郑… ②黄… III. ①程序语言—程序
设计 IV. ①TP312

中国版本图书馆CIP数据核字(2017)第264241号

版权声明

Original English language edition, entitled *Go Web Programming* by Sau Sheong Chang published by Manning Publications Co., 209 Bruce Park Avenue, Greenwich, CT 06830. Copyright © 2016 by Manning Publications Co. Simplified Chinese-language edition copyright © 2017 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由 Manning Publications Co.授权人民邮电出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

-
- ◆ 著 [新加坡] 郑兆雄 (Sau Sheong Chang)
 - 译 黄健宏
 - 责任编辑 杨海玲
 - 责任印制 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京中新伟业印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 19.5
 - 字数: 425 千字 2017 年 12 月第 1 版
 - 印数: 1—5 000 册 2017 年 12 月北京第 1 次印刷
 - 著作权合同登记号 图字: 01-2016-6529 号
-

定价: 79.00 元

读者服务热线: (010) 81055410 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

广告经营许可证: 京东工商广登字 20170147 号

内容提要

本书全面介绍使用 Go 语言开发 Web 应用所需的全部基本概念，并详细讲解如何运用现代设计原则使用 Go 语言构建 Web 应用。本书通过大量的实例介绍核心概念（如处理请求和发送响应、模板引擎和数据持久化），并深入讨论更多高级主题（如并发、Web 应用程序测试以及部署到标准系统服务器和 PaaS 提供商）。

本书以一个网络论坛为例，讲解如何使用请求处理器、多路复用器、模板引擎、存储系统等核心组件构建一个 Go Web 应用，然后在这一应用的基础上，构建出相应的 Web 服务。值得一提的是，本书在介绍 Go Web 开发方法时，基本上只用到 Go 语言自带的标准库，而不会用到任何特定的 Web 框架，读者学到的知识将不会局限于特定的框架，即使将来需要用到现成的框架或者自行构建框架，仍然会从本书中获益。本书除了讲解具体的 Web 开发方法，还介绍如何对 Go Web 应用进行测试，如何使用 Go 的并发特性提高 Web 应用的性能，以及如何在 Heroku、Google App Engine、Digital Ocean 等云平台上部署 Go Web 应用；此外，书中还传授一些 Go Web 开发方面的经验和提示。这些重要的实践知识将帮助读者快速成为真正具有生产力的 Go Web 开发者。

阅读本书需要读者具备基本的 Go 语言编程技能并掌握 Go 语言的语法。本书适合所有想用 Go 语言进行 Web 开发的读者阅读，无论是 Web 开发的初学者还是入行已久的开发者都会在阅读本书的过程中有所收获。

译者记事

随着近年来 Web 开发的盛行，很多相关书籍也随之如雨后春笋般出现，然而在这些书籍当中，绝大多数书籍都只关注表面的实现代码，而对代码背后的技术原理却少有提及。读者在看这类书籍时，虽然可以学到某个框架或者某个库的 API，并根据书中给出的代码搭建出一个个演示程序（demo），但是对隐藏在这些代码之下的原理却一无所知。这种停留在表面的理解一旦离开了书本的指导，就会让人感到寸步难行，不知所措。

本书的独特之处在于，它抛开了现有的所有 Go Web 框架，仅仅通过 Go 语言内置的标准库来展示如何去构建一个 Web 应用或 Web 服务。这样做的好处是，无论将来读者是使用这些标准库来构建 Web 应用，还是使用现成的框架去构建 Web 应用，又或者使用自己建造的框架去构建 Web 应用，本书介绍的知识都是非常有用的：如果使用的是现成的框架，那么这些框架的内部实现通常就是由本书介绍的 Go 标准库构建的；如果选择自建框架，那么将有很大概率会用到本书介绍的 Go 标准库。因此，不论在何种情况下，本书对于构建 Go Web 应用都是非常有帮助的。

本书的另一个优点是，它在介绍 Web 应用开发技术的同时，也介绍了隐藏在这些技术背后的基础知识。比如，在介绍 Web 处理器（handler）的创建方法之前，本书就先深入浅出地介绍了 HTTP 协议，然后才说明具体的请求处理方法以及响应返回方法；又比如说，在介绍会话（session）技术时，本书就先说明了 HTTP 协议的无状态性质，然后才说明如何使用会话去解决这一问题；类似的例子在书里面还有很多，不一而足。对刚开始接触 Web 开发的读者来说，本书这种“知其然，也知其所以然”的教授方式能够让读者打好 Web 开发的基础，从而达到事半功倍的效果；此外，对那些已经有一定 Web 开发经验的读者来说，本书将在介绍 Go Web 开发方法的同时，帮助读者回顾和巩固 Web 开发的相关基础知识，并藉此成为更好的 Web 开发者。

综上所述，我认为这本书对所有关心 Web 开发的人来说，都是非常值得一读的——无论读者使用的是 Go 语言还是其他语言、X 框架还是 Y 框架，无论读者是 Web 开发的初学者还是入行已久的开发者，应该都会在阅读本书的过程中有所收获。

关于本书的翻译

这本《Go Web 编程》是我的第二部译作，在翻译第一部译作《Redis 实战》的时候，因为受经验、知识以及时间等条件限制，我只能把时间尽量花在保证译文的准确性上，但是对于译文本身的可读性却未能有太多的关注。这次在翻译这本《Go Web 编程》的过程中，我给自己订立了更高的目标，那就是，在保证译文正确性的前提下，通过合理的用词遣句，让译文更符合中文表达方式，并且更具表现力。

以本书的前言原文为例，其中就有一句“My own journey in developing applications for the web started around the same time, in the mid-1990s”，这句话的原意是说作者的 Web 开发生涯跟万维网的发展轨迹正好重合，因此把它单纯地译为“本人的 Web 应用开发生涯也是从 20 世纪 90 年代中期开始……”是完全没有问题的，但是通过在句子前面添加“无独有偶”一词来与“around the same time”的翻译“也是”相互呼应，就会一下子给译文带来画龙点睛的效果：“无独有偶，本人的 Web 应用开发生涯也是从 20 世纪 90 年代中期开始……”

继续以前言为例，在这篇文章的原文当中，出现了不少常见的英文短语和词汇，这些短语和词汇通常都有一个正确、常见并且平庸的翻译，但是本书却抛弃了这些翻译，转而选择了更准确也更有表现力的译法。比如说，“Writing web applications has changed dramatically over the years”中的“changed dramatically”没有直译为“发生了戏剧性的变化”，而是翻译为“发生了翻天覆地的变化”；“Almost as soon as the first web applications were written, web application frameworks appeared”中的“were written”和“appeared”没有直译为“被编写出来之后”以及“出现”，而是分别翻译为“闪亮登场”和“应运而生”，前者突出了 Web 应用的出现对于互联网的巨大改变，而后者则突出了 Web 应用和 Web 框架之间相辅相成的关系。类似的例子还有很多很多，并且它们不仅出现在了前言里，还出现在了本书的正文当中。

当然，提高译文的可读性并不是一件一蹴而就的事。为了让译文更有“中文味”，本书的大多数译文都已三易其稿，有时候仅仅为了挑选出一个更恰当的词语或成语，就不得不对着词典推敲半天。这本书的翻译从 2016 年 8 月开始，到 2017 年 8 月交稿，整整跨越了一年时间，其中翻译原文和润色译文两项工作花费的时间可谓各占一半。如果读者能够从译文的字里行间感受到这种润物细无声的优化，那将是对本人翻译工作最好的肯定。

另外，因为这是一本使用 Go 语言标准库进行 Web 开发的书，所以对 Go Web 开发相关标准库的理解程度将是能否准确地翻译本书技术内容的关键。为了进一步熟悉本书用到的标准库，本人通读了书中用到的各个标准库的文档，阅读了其中部分标准库的源码，并且因为有时候“好记性比不上烂笔头”，所以本人还翻译了其中一部分标准库文档，力求在尽可能掌握标准库细节的情况下，再进行翻译，尽量做到知其然也知其所以然，而不是单纯地根据纸面上的文字和代码进行翻译。

最后，在翻译本书的过程中，本人也发现了原著中大大小小数十个 bug，并在译文中一一进行了修正。综上所述，读者看到的这个译本从某个角度来说将比原著更准确也更易读。这也是我

一直以来在实践翻译工作时的信念——译作不应该是原著的“劣化版”，而是应该以“青出于蓝而胜于蓝”的方式超越原著。当然，要做到这一点并不是一件容易的事，但每一个合格的译者都应该以此为目标，不断奋斗。

读者服务网站

为了更好地服务本书读者，本人专门为本书搭建了读者服务网站 <http://gwpcn.com>。读者只要访问这个网站，就可以查看到与本书有关的各项信息，如本书的简介、目录、试读内容、作译者介绍、勘误信息、购买地址以及源代码下载地址等。

除此之外，正如之前所说，本人在翻译本书的过程中也翻译了一部分 Go 标准库的文档，这些文档可以通过地址 <http://cngolib.com> 查看。

致谢

感谢人民邮电出版社以及杨海玲编辑对我的信任，将这样一本有趣而且重要的书交给我翻译，我衷心希望这本译作能够取得好成绩，从而不辜负他们对我的信任与期待。

感谢网络上一直关心这个译本出版进度的读者朋友们，他们的支持和鼓励让我不敢有所怠慢，争取竭尽自己所能，贡献出一个能够为大家所喜爱的译本。

最后也是最重要的，我要感谢本书翻译过程中一如既往地全力支持我的家人和朋友，多亏了他们的帮助，本书的翻译工作才得以顺利完成。

黄健宏
2017 年秋

译者简介



黄健宏 (huangz)，一位 1990 年出生的计算机技术图书作译者，《Redis 设计与实现》一书的作者，《Redis 实战》一书的译者。

除了已出版的两本作品之外，他还创作和翻译了《Go 标准库中文文档》《Redis 命令参考》《SICP 解题集》等一系列开源文档。

要了解关于黄健宏的更多信息，请访问他的个人主页 <http://huangz.me>。

前言

自互联网从 20 世纪 90 年代中期诞生以来，Web 应用就以这样或那样的方式存在了。虽然 Web 应用在最初只能传输静态网页，但它很快就升级和演变成了一个令人眼花缭乱、能够传输各种数据以及实现各种功能的动态系统。无独有偶，本人也是从 20 世纪 90 年代中期开始接触 Web 应用开发的，在迄今为止的职业生涯当中，我把大部分时间都花费在了大规模 Web 应用的设计、开发以及团队管理上面，并且在这期间还使用过多种不同的编程语言和框架，其中包括 Java、Ruby、Node.js、PHP、Perl、Elixir 甚至是 Smalltalk。

几年前，我因为一次偶然的机会接触到了 Go 语言，并迅速被它的简单和清爽直率所吸引，而当我意识到只使用 Go 的标准库就可以快速地构建完整、高效并且可扩展的 Web 应用和服务时，我对 Go 的喜爱又更进了一步。使用 Go 语言编写的代码不仅易懂、直截了当，而且还能够快速、简单地编译成一个独立的可部署二进制文件。更关键的是，我不必投入大量服务器就可以让自己的 Web 应用变得可扩展且具备生产能力。很自然地，所有的这些优点都使 Go 成为了我在 Web 应用开发方面最新的心头好语言。

从当初传输静态内容到现在通过 HTTP 传输动态数据，从当初使用服务器传输 HTML 内容，到现在使用客户端单页应用去处理通过 HTTP 传输的 JSON 数据，Web 应用的开发方式已经发生了翻天覆地的变化。几乎就在 Web 应用闪亮登场的同时，Web 框架也应运而生，并使程序员可以更为容易地去开发 Web 应用。这二十多年以来，绝大多数编程语言都会有至少一个 Web 应用框架，其中很多语言甚至会有一大堆框架可用，而当今出现的绝大多数应用都是 Web 应用。

尽管 Web 应用框架的风靡使开发 Web 应用变得更加容易了，但这些框架在带来方便的同时也隐藏了大量的细节——Web 应用开发者对于万维网的运作方式知之甚少甚至一窍不通，这样的情况正在变得越来越常见。幸运的是，通过 Go 语言，我发现了一种正确地教授 Web 应用开发基础知识的绝佳工具，它能够让 Web 应用开发重新回到简单直接的状态：程序需要考虑的就是如何处理 HTTP 协议，以及如何通过 HTTP 协议传输内容和数据，并且满足这两个要求只需要用到 Go 语言本身提供的工具——不需要用到外部库，也不需要用到外部的依赖。

在拿定注意之后，我就向 Manning 出版社提交了一个撰写 Go 语言编程书籍的构思，这个

构思关注的是如何在只使用标准库的情况下，向读者传授从零开始构建 Web 应用的方法，而 Manning 出版社也很快就同意了我的构思并开启了这个项目。尽管本书的撰写工作持续了一段时间才得以完成，但是在写作的过程中，抢先预览版带来的反馈总是不断地鼓舞着我。最后，我希望读者能够像我享受创作本书的过程一样，享受阅读本书的过程，并且在这个过程中能够有所收获。

致谢

本书最初的想法是在只使用标准库的情况下教授基本的 Go Web 编程知识。说实在的，刚开始的时候我并不确定这个想法是否能够行得通，但那些花费自己血汗钱来购买本书抢先预览版的读者给了我鼓励和动力来实现这个想法，因此在这里我要向我的读者们致以诚挚的感谢！

写书是一项团队协作活动，尽管本书的封面上只记载了我一个人的名字，但实际上大量幕后人员也为这本书付出了自己的心血，他们分别是：

- Marina Michaels，来自地球另一侧的一位勤劳且高效的编辑，她总是不知疲倦地配合我的工作，并且为了我们之间巨大的时差而不断地调整自己的日程表；
- Manning 出版社的相关工作人员：文字编辑 Liz Welch 和校对 Elizabeth Martin，他们的火眼金睛让错误无处可藏，负责营销和推广本书的 Candace Gillhoolley 和 Ana Romac，以及将我的原稿变为本书的 Kevin Sullivan 和 Janet Vail；
- Jimmy Frasc 对我的原稿进行了一次完整的技术校对，而我的审稿人 Alex Jacinto、Alexander Schwartz、Benoit Benedetti、Brian Cooksey、Doug Sparling、Ferdinando Santacroce、Gualtiero Testa、Harry Shaun Lippy、James Tyo、Jeff Lim、Lee Brandt、Mike Bright、Quintin Smith、Rebecca Jones、Ryan Pulling、Sam Zaydel 和 Wes Shaddix 则在撰写原稿的 4 个阶段中为我提供了大量有价值的反馈；
- 这本书的抢先预览版一经释出，我在新加坡 Go 社区的朋友们就迫不及待地把它向全世界广而告之了，特别值得一提的是 Kai Hendry，他为本书制作了一个详细的评论视频。

另外，我还要感谢 Go 的创造者 Robert Griesemer、Rob Pike 和 Ken Thompson，以及 net/http、html/template 等 Web 标准库的开发者，特别是 Brad Fitzpatrick，没有他们的辛勤付出，这本书就不可能出现。

最后，也是最必不可少的，我要感谢我的家人，包括我亲爱的妻子 Wooi Ying，以及在身高方面后来居上的我的儿子 Kai Wen。我希望自己能够通过创作这本书给他带来启发，我也希望他会自豪地阅读这本书，并从中有所收获。

关于本书

本书将完整地介绍使用 Go 语言开发 Web 应用所需的全部基本概念,并且在这个过程中只使用 Go 语言自带的标准库。尽管本书的部分章节会对其他库以及其他主题进行讨论,比如如何测试 Web 应用以及如何部署 Web 应用,但本书的主要目的还是教读者如何在只使用 Go 标准库的情况下进行 Web 开发。

本书要求读者具备基本的 Go 编程技能并掌握 Go 语言的语法。如果读者不具备这些知识,可以阅读由 William Kennedy、Brian Ketelsen 和 Erik St. Martin 创作的 *Go in Action*^① 一书,该书也是由 Manning 出版社出版的。由 Addison-Wesley 出版社出版、Alan Donovan 和 Brian Kernighan 创作的 *The Go Programming Language*^② 也是一本值得一读的好书。除了以上提到的两本书之外,网上也有非常多免费的 Go 教程可供浏览,比如,Go 官方网站的《Go 入门教程》(A Tour of Go) (<http://tour.golang.org/>) 就是一个很棒的例子。

内容编排

本书由 10 章和一个附录组成。

第 1 章会介绍使用 Go 开发 Web 应用的方法,并阐述这种做法的优点所在。除此之外,本章还会对 HTTP 协议等构成 Web 应用的关键概念做深入浅出的介绍。

第 2 章会以一步一个脚印的方式,带领读者去构建一个简单的网上论坛,以此来向读者展示如何使用 Go 构建一个典型的 Web 应用。

第 3 章会更加详细地展示使用 net/http 包接收 HTTP 请求的方法。读者将学会如何编写 Go Web 服务器监听 HTTP 请求,以及如何使用处理器和处理器函数处理这些请求。

第 4 章会继续介绍处理 HTTP 请求的相关细节,重点讲述 Go 是如何处理请求并返回响应的。

① *Go in Action* 的中文版已由人民邮电出版社出版,中文版书名为《Go 语言实战》。——译者注

② *The Go Programming Language* 的中文版已由机械工业出版社出版,中文版书名为《Go 程序设计语言》。——译者注

除此之外，读者还将学会如何从 HTML 表单中获取数据以及如何使用 cookie。

第 5 章将会介绍由 text/template 库和 html/template 库组成的 Go 模板引擎。读者将会看到 Go 提供的各种模板机制，并学会如何使用 Go 的布局（layout）。

第 6 章将会对 Go 的存储策略进行讨论。读者将学会如何通过结构将数据存储到内存里面，如何通过 CSV 格式以及 gob 二进制格式将数据存储到文件系统里面，以及如何通过 SQL 和 SQL 映射器去访问关系数据库。

第 7 章将展示使用 Go 语言构建 Web 服务的方法。读者不仅会学到如何使用 Go 语言构建一个简单的 Web 服务，还会学到如何使用 Go 语言创建并分析 XML 数据和 JSON 数据。

第 8 章将向读者传授在不同层级中测试 Go Web 应用的不同方法，其中包括单元测试、基准测试以及 HTTP 测试；除此之外，这一章还会简单介绍几个第三方测试库。

第 9 章会介绍在 Web 应用中使用 Go 语言的并发特性的方法。读者将会了解到 Go 语言的各个并发特性，并学会如何使用这些特性提高一个图像生成 Web 应用的性能。

第 10 章是本书的最后一章，它将展示 Go Web 应用的部署方法。读者将会学到如何把应用部署到独立的服务器上，如何把应用部署到 Heroku、Google App Engine 之类的云平台上，以及如何把应用部署到 Docker 容器里面。

最后，本书的附录会展示在不同平台上安装和设置 Go 环境的方法。

代码的约定以及下载

本书通过代码清单以及正文内嵌的方式展示了大量源代码。为了跟一般的正文区别开来，书中的源代码都会使用等宽字体。为了凸显某些代码在不同章节之间的区别，又或者为了强调正文中讨论的某些代码，本书有时候也会以加粗的方式显示代码。

除此之外，本书的电子书还会使用彩色字体来凸显代码命令以及代码输出：

```
curl -i 127.0.0.1:8080/write
HTTP/1.1 200 OK
Date: Tue, 13 Jan 2015 16:16:13 GMT
Content-Length: 95
Content-Type: text/html; charset=utf-8

<html>
<head><title>Go Web Programming</title></head>
<body><h1>Hello World</h1></body>
</html>
```

本书展示的所有代码都可以在 www.manning.com/books/go-web-programming 和 github.com/sausheong/gwp 找到^①。

① 本书展示的所有代码也可以在异步社区（www.epubit.com.cn）中本书页面免费下载。——编者注

作者简介



郑兆雄 (Sau Sheong Chang), 现任新加坡能源有限公司数字技术总裁, 在此之前他曾经担任过 PayPal 的消费者工程经理。Sau 是 Ruby 社区和 Go 社区一位活跃的贡献者, 除了创作书籍之外, 他还为开源项目提交代码, 并在各种技术研讨会和技术会议上发言。

作者在线论坛

购买本书英文版的读者可以免费地访问由 Manning 出版社开设的私有 Web 论坛, 可以在论坛里面撰写书评、提出技术问题并接受来自作者和其他读者的帮助。为了访问并订阅论坛, 需要先使用浏览器访问 www.manning.com/books/go-web-programming, 这个页面会告诉读者注册账号和访问论坛的方法, 除此之外, 该页面还列举了论坛提供的各种帮助以及论坛的各项规章制度。

Manning 出版社承诺为读者提供论坛作为场所, 以便读者之间以及读者和作者之间可以进行有意义的对话, 但 Manning 并不保证作者的参与程度——作者对论坛的任何贡献都是自愿并且无偿的, 因此读者应该尽可能地提出一些具有挑战性的问题以便引起作者的兴趣。

只要本书仍在正常销售, 本书的作者在线论坛以及论坛上已有的帖子就会一直可供访问。

关于封面插图

本书的封面插图系 Paolo Mercuri (1804—1884) 所作，标题为“穿着中世纪服装的男人”，该插图来源于 Camille Bonnard 搜集并编辑的 *Costumes Historiques* (服装史) 多卷本，该书于 19 世纪 50 或 60 年代在巴黎出版，它搜集了大量 12 世纪、13 世纪、14 世纪和 15 世纪的历史服装。随着异国风情和历史文化在 19 世纪风靡，人们开始着迷于这类服装收藏本，并藉此去探索自己所在的世界以及已经远去的旧世界。

在这一历史画册中，Mercuri 丰富多彩的画作让我们生动地回想起了数百年前，世界各地不同城市和地区之间的文化差异。无论是在街道还是乡间，仅仅通过人们的着装就可以八九不离十地辨识他们的社会地位、从事的行业和职业。在经历了数个世纪的变迁以后，人们的着装方式已经发生了很大的变化，当初丰富多彩的地区多样性也已逐渐消失。时至今日，仅仅通过着装已经很难区分不同大洲的居民了，更别说想要知道他们所在的国家 and 城市、知悉他们的社会地位和职业了。乐观地讲，也许我们已经放弃了追求文化上的多样性，转为拥抱更丰富多彩也更快节奏的技术生活了。

在计算机书籍正在变得越来越相似、越来越同质化的今天，Manning 出版社希望通过 Mercuri 的作品，将数个世纪以前丰富多彩的地区生活融入图书封面，以此来赞美计算机行业不断创新和敢为人先的精神。

目录

第一部分 Go 与 Web 应用

第 1 章 Go 与 Web 应用 3

- 1.1 使用 Go 语言构建 Web 应用 3
 - 1.1.1 Go 与可扩展 Web 应用 4
 - 1.1.2 Go 与模块化 Web 应用 4
 - 1.1.3 Go 与可维护的 Web 应用 5
 - 1.1.4 Go 与高性能 Web 应用 5
- 1.2 Web 应用的工作原理 6
- 1.3 HTTP 简介 7
- 1.4 Web 应用的诞生 8
- 1.5 HTTP 请求 9
 - 1.5.1 请求方法 10
 - 1.5.2 安全的请求方法 11
 - 1.5.3 幂等的请求方法 11
 - 1.5.4 浏览器对请求方法的支持 11
 - 1.5.5 请求首部 12
- 1.6 HTTP 响应 13
 - 1.6.1 响应状态码 13
 - 1.6.2 响应首部 14
- 1.7 URI 15
- 1.8 HTTP/2 简介 16
- 1.9 Web 应用的各个组成部分 16

第 2 章 ChitChat 论坛 23

- 1.9.1 处理器 17
- 1.9.2 模板引擎 18
- 1.10 Hello Go 18
- 1.11 小结 21
- 2.1 ChitChat 简介 23
- 2.2 应用设计 24
- 2.3 数据模型 26
- 2.4 请求的接收与处理 27
 - 2.4.1 多路复用器 27
 - 2.4.2 服务静态文件 29
 - 2.4.3 创建处理器函数 29
 - 2.4.4 使用 cookie 进行访问控制 30
- 2.5 使用模板生成 HTML 响应 33
- 2.6 安装 PostgreSQL 38
 - 2.6.1 在 Linux 或 FreeBSD 系统上安装 38
 - 2.6.2 在 Mac OS X 系统上安装 39
 - 2.6.3 在 Windows 系统上安装 39
- 2.7 连接数据库 39
- 2.8 启动服务器 44
- 2.9 Web 应用运作流程回顾 45
- 2.10 小结 46

第二部分 Web 应用的基本组成部分

第3章 接收请求 49

- 3.1 Go 的 net/http 标准库 49
- 3.2 使用 Go 构建服务器 51
 - 3.2.1 Go Web 服务器 51
 - 3.2.2 通过 HTTPS 提供服务 53
- 3.3 处理器和处理器函数 56
 - 3.3.1 处理请求 56
 - 3.3.2 使用多个处理器 58
 - 3.3.3 处理器函数 59
 - 3.3.4 串联多个处理器和处理器函数 61
 - 3.3.5 ServeMux 和 DefaultServeMux 65
 - 3.3.6 使用其他多路复用器 66
- 3.4 使用 HTTP/2 68
- 3.5 小结 70

第4章 处理请求 72

- 4.1 请求和响应 72
 - 4.1.1 Request 结构 73
 - 4.1.2 请求 URL 73
 - 4.1.3 请求首部 74
 - 4.1.4 请求主体 76
- 4.2 Go 与 HTML 表单 77
 - 4.2.1 Form 字段 79
 - 4.2.2 PostForm 字段 80
 - 4.2.3 MultipartForm 字段 81
 - 4.2.4 文件 83
 - 4.2.5 处理带有 JSON 主体的 POST 请求 85
- 4.3 ResponseWriter 86
- 4.4 cookie 91
 - 4.4.1 Go 与 cookie 91
 - 4.4.2 将 cookie 发送至浏览器 92
 - 4.4.3 从浏览器里面获取 cookie 94
 - 4.4.4 使用 cookie 实现闪现消息 96
- 4.5 小结 99

第5章 内容展示 100

- 5.1 模板引擎 100
- 5.2 Go 的模板引擎 102

- 5.2.1 对模板进行语法分析 104
- 5.2.2 执行模板 105

5.3 动作 106

- 5.3.1 条件动作 106
- 5.3.2 迭代动作 108
- 5.3.3 设置动作 109
- 5.3.4 包含动作 111

5.4 参数、变量和管道 113

5.5 函数 114

5.6 上下文感知 116

- 5.6.1 防御 XSS 攻击 119
- 5.6.2 不对 HTML 进行转义 121

5.7 嵌套模板 122

5.8 通过块动作定义默认模板 126

5.9 小结 127

第6章 存储数据 128

- 6.1 内存存储 128
- 6.2 文件存储 131
 - 6.2.1 读取和写入 CSV 文件 133
 - 6.2.2 gob 包 135
- 6.3 Go 与 SQL 137
 - 6.3.1 设置数据库 138
 - 6.3.2 连接数据库 140
 - 6.3.3 创建帖子 142
 - 6.3.4 获取帖子 144
 - 6.3.5 更新帖子 145
 - 6.3.6 删除帖子 145
 - 6.3.7 一次获取多篇帖子 146
- 6.4 Go 与 SQL 的关系 147
 - 6.4.1 设置数据库 147
 - 6.4.2 一对多关系 150
- 6.5 Go 与关系映射器 152
 - 6.5.1 Sqlx 152
 - 6.5.2 Gorm 154
- 6.6 小结 157

第三部分 实战演练

第7章 Go Web 服务 161

- 7.1 Web 服务简介 161
- 7.2 基于 SOAP 的 Web 服务简介 163