

# 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 Frasché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*<sup>①</sup> 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*<sup>②</sup> 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](http://www.manning.com/books/go-web-programming) 和 [github.com/sausheong/gwp](https://github.com/sausheong/gwp) 找到<sup>①</sup>。

---

<sup>①</sup> 本书展示的所有代码也可以在异步社区（[www.epubit.com.cn](http://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](http://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明在 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

- 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

### 第 2 章 ChitChat 论坛 23

- 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

3

### 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

4

### 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

5

### 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

6

### 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

7

### 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

## 7.3 基于 REST 的 Web 服务

### 简介 166

#### 7.3.1 将动作转换为资源 168

#### 7.3.2 将动作转换为资源的属性 169

## 7.4 通过 Go 分析和创建

### XML 169

#### 7.4.1 分析 XML 169

#### 7.4.2 创建 XML 177

## 7.5 通过 Go 分析和创建

### JSON 180

#### 7.5.1 分析 JSON 181

#### 7.5.2 创建 JSON 184

## 7.6 创建 Go Web 服务 187

## 7.7 小结 194

# 8 第 8 章 应用测试 196

## 8.1 Go 与测试 196

## 8.2 使用 Go 进行单元测试 197

### 8.2.1 跳过测试用例 201

### 8.2.2 以并行方式运行测试 202

### 8.2.3 基准测试 203

## 8.3 使用 Go 进行 HTTP 测试 206

## 8.4 测试替身以及依赖注入 210

## 8.5 第三方 Go 测试库 217

### 8.5.1 Gocheck 测试包简介 217

### 8.5.2 Ginkgo 测试框架简介 223

## 8.6 小结 228

# 9 第 9 章 发挥 Go 的并发优势 230

## 9.1 并发与并行的区别 230

## 9.2 goroutine 232

### 9.2.1 使用 goroutine 232

### 9.2.2 goroutine 与性能 235

### 9.2.3 等待 goroutine 238

## 9.3 通道 239

### 9.3.1 通过通道实现同步 240

### 9.3.2 通过通道实现消息传递 242

### 9.3.3 有缓冲通道 243

### 9.3.4 从多个通道中选择 244

## 9.4 在 Web 应用中使用并发 247

### 9.4.1 创建马赛克图片 248

### 9.4.2 马赛克图片 Web 应用 251

### 9.4.3 并发版马赛克图片生成 Web 应用 254

## 9.5 小结 262

# 10 第 10 章 Go 的部署 263

## 10.1 将应用部署到独立的服务器 264

## 10.2 将应用部署到 Heroku 270

## 10.3 将应用部署到 Google App Engine 273

## 10.4 将应用部署到 Docker 278

### 10.4.1 什么是 Docker 278

### 10.4.2 安装 Docker 279

### 10.4.3 Docker 的理念与组件 280

### 10.4.4 Docker 化一个 Go Web 应用 281

### 10.4.5 将 Docker 容器推送至互联网 283

## 10.5 部署方法之间的对比 286

## 10.6 小结 287

## 附录 安装和设置 Go 288



## 第一部分

# Go 与 Web 应用

Web 应用是当今使用最为广泛的一类软件应用，连接至互联网的人们基本上每天都在使用 Web 应用。因为很多看上去像是原生应用的移动应用都在内部包含了使用 Web 技术构建的组件，所以使用移动设备的人们实际上也是在使用 Web 应用。

因为编写 Web 应用必须对 HTTP 有所了解，所以接下来的两章将对 HTTP 进行介绍。除此之外，我们还会了解到使用 Go 语言编写 Web 应用的优点，并且实际使用 Go 语言来构建一个简单的网上论坛，然后鸟瞰 Web 应用的各个组成部分。

# 第 1 章 Go 与 Web 应用

## 本章主要内容

- Web 应用的定义
- 使用 Go 语言编写 Web 应用的优点
- Web 应用编程的基本知识
- 使用 Go 语言编写一个极为简单的 Web 应用

Web 应用在我们的生活中无处不在。看看我们日常使用的各个应用程序，它们要么是 Web 应用，要么是移动 App 这类 Web 应用的变种。无论哪一种编程语言，只要它能够开发出与人类交互的软件，它就必然会支持 Web 应用开发。对一门崭新的编程语言来说，它的开发者首先要做的一件事，就是构建与互联网（internet）和万维网（World Wide Web）交互的库（library）和框架，而那些更为成熟的编程语言还会有各种五花八门的 Web 开发工具。

Go 是一门刚开始崭露头角的语言，它是为了让人们能够简单且高效地编写后端系统（back end system）而创建的。这门语言拥有众多先进的特性，并且密切关注程序员的生产力以及各种与速度相关的事项。和其他语言一样，Go 语言也提供了对 Web 编程的支持。自从问世以来，Go 语言在编写 Web 应用以及“x 即服务系统”（\*as-a-service system）方面就受到了热烈追捧。

本章接下来将列举一些使用 Go 编写 Web 应用的优点，并介绍一些关于 Web 应用的基本知识。

## 1.1 使用 Go 语言构建 Web 应用

“为什么要使用 Go 语言编写 Web 应用呢？”作为本书的读者，我想你肯定很想知道这个问题的答案。本书是一本教人们如何使用 Go 语言进行 Web 编程的图书，而作为本书的作者，我的任务就是向你解释为什么人们会使用 Go 语言进行 Web 编程。本书将在接下来的内容中陆续介绍 Go 语言在 Web 开发方面的优点，我衷心地希望你也能够对这些优点有感同身受的想法。

Go 是一门相对比较年轻的编程语言，它拥有繁荣并且仍在不断成长的社区，并且它也非常