EXPERT INSIGHT

Golang для профи

Создаем профессиональные утилиты, параллельные серверы и сервисы

Третье издание

Михалис Цукалос

Packt

Mastering Go

Third Edition

Harness the power of Go to build professional utilities and concurrent servers and services

Mihalis Tsoukalos

Packt›

BIRMINGHAM - MUMBAI

Golang для профи

Третье издание

Создаем профессиональные утилиты,
параллельные серверы и сервисы

Михалис Цукалос

ПИТЕР®

Санкт-Петербург • Москва • Минск

2024

ББК 32.988.02-018.1
УДК 004.43
Ц85

Цукалос Михалис

Ц85 Golang для профи: Создаем профессиональные утилиты, параллельные серверы и сервисы. 3-е изд. — СПб.: Питер, 2024. — 624 с.: ил. — (Серия «Для профессионалов»).

ISBN 978-5-4461-1999-8

Язык Go — это простой и понятный язык для создания высокопроизводительных систем будущего. Используйте Go в реальных производственных системах. В новое издание включены такие темы, как создание серверов и клиентов RESTful, знакомство с дженериками Go и разработка серверов и клиентов gRPC.

Третье издание «Golang для профи» исследует практические возможности Go и описывает такие продвинутые темы, как параллелизм и работа сборщика мусора Go, использование Go с Docker, разработка мощных утилит командной строки, обработка данных в формате JSON (JavaScript Object Notation) и взаимодействие с базами данных. Кроме того, книга дает дополнительные сведения о работе внутренних механизмов Go, знание которых позволит оптимизировать код на Go и использовать типы и структуры данных новыми и необычными способами.

Также охватываются некоторые нюансы и идиомы языка Go, предлагаются упражнения и приводятся ссылки на ресурсы для закрепления полученных знаний.

Станьте опытным программистом на Go, создавая системы и внедряя передовые методы программирования на Go в свои проекты!

16+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ББК 32.988.02-018.1
УДК 004.43

ISBN 978-1801079310 англ.

ISBN 978-5-4461-1999-8

Краткое содержание

Оглавление

Об авторе

Михалис Цукалос — системный инженер UNIX, который увлекается написанием технических текстов. Автор книг *Go Systems Programming* и *Mastering Go*[1], как первого, так и второго изданий. Получил степень бакалавра математики в Университете Патр и степень магистра информационных технологий в Университетском колледже Лондона. Написал более 300 технических статей для различных журналов, включая *Sys Admin*, *MacTech*, *Linux User and Developer*, *Usenix ;login:*, *Linux Format* и *Linux Journal*. В круг научных интересов Михалиса входят временные ряды, базы данных и индексирование.

Вы можете связаться с автором по адресу https://www.mtsoukalos.eu/ и @mactsouk.

Поскольку написание книги — командная работа, я хотел бы поблагодарить за помощь сотрудников издательства Packt Publishing. Спасибо Амиту Рамадасу за ответы на все мои вопросы, Шайлешу Джайну — за то, что убедил начать работу над третьим изданием Mastering Go, Эдварду Докси — за полезные комментарии и предложения и Дереку Паркеру — за его хорошую работу.

Наконец, я хотел бы поблагодарить вас, читателя, за выбор этой книги. Надеюсь, она окажется вам полезной.

---

[1] *Цукалос М.* Golang для профи. Работа с сетью, многопоточность, структуры данных и машинное обучение с Go. — СПб.: Питер, 2020.

О научном редакторе

Дерек Паркер — инженер-программист в Red Hat. Создатель отладчика Delve для Go и автор компилятора Go, компоновщика и стандартной библиотеки. Является автором проектов с открытым исходным кодом и специалистом по сопровождению ПО, работал над множеством проектов — от интерфейсного JavaScript до низкоуровневого ассемблерного кода.

Я хотел бы поблагодарить мою жену Эрику и двух наших замечательных детей за то, что они дали мне возможность работать над этим проектом.

Предисловие

Книга, которую вы сейчас читаете, называется «Golang для профи: Создаем профессиональные утилиты, параллельные серверы и сервисы» (3-е издание), и она вся о том, как стать лучшим разработчиком на Go! Если у вас есть второе издание, то не выбрасывайте его — Go не так уж сильно изменился, и второе издание по-прежнему полезно. Однако третье издание во многих аспектах лучше!

Добавлено много интересных новых тем, включая написание сервисов RESTful, работу с протоколом WebSocket и использование GitHub Actions и GitLab Actions для проектов Go, а также совершенно новая глава о дженериках и разработке множества полезных утилит. Кроме того, я постарался сделать это издание меньше, чем второе, и улучшить структуру книги, чтобы облегчить вам ее чтение и ускорить его.

Я также постарался включить нужное количество теории и практики — но только вы, читатель, можете сказать, насколько мне это удалось! Попробуйте выполнить упражнения в конце каждой главы и не стесняйтесь обращаться ко мне с идеями, которые могут еще больше улучшить будущие издания этой книги!

Для кого эта книга

Книга предназначена для программистов на Go среднего уровня, которые хотят улучшить свои навыки и перевести их на новый уровень. Она также будет полезна опытным разработчикам на других языках программирования, которые хотят изучить Go, не углубляясь в основы программирования.

Структура издания

Глава 1 начинается с рассказа об истории возникновения, важных свойствах и преимуществах Go. После этого описываются утилиты `godoc` и `go doc` и объясняется, как компилировать и исполнять программы на Go. Далее в главе рассказывается о выводе данных и получении пользовательского ввода, работе

с аргументами командной строки и использовании файлов журнала. Наконец, мы разработаем базовую версию приложения для телефонной книги, которую будем совершенствовать в следующих главах.

В главе 2 рассматриваются основные типы данных Go, как числовые, так и нечисловые, а также массивы и срезы, позволяющие группировать данные одного типа. В ней также рассматриваются указатели Go, константы и работа с датами и временем. Последняя часть главы посвящена генерации случайных чисел и заполнению приложения телефонной книги случайными данными.

Глава 3 начинается с карт, после чего переходит к структурам и ключевому слову `struct`. Кроме того, в ней вы найдете информацию о регулярных выражениях, сопоставлении с образцами и работе с CSV-файлами. Наконец, мы улучшим приложение телефонной книги, добавив в него постоянное хранение данных.

Глава 4 посвящена рефлексии, интерфейсам и методам типов — функциям, прикрепленным к типам данных. В ней также описывается использование интерфейса `sort.Interface` для сортировки фрагментов, использование пустого интерфейса, а также представлены утверждения типа, переключатели типа и тип данных `error`. Дополнительно, прежде чем улучшать приложение телефонной книги, мы обсудим, как Go может имитировать некоторые объектно-ориентированные концепции.

Глава 5 посвящена пакетам, модулям и функциям, которые являются основными элементами пакетов. Среди прочего мы создадим пакет Go, позволяющий взаимодействовать с базой данных PostgreSQL, документацию для него, а также объясним не всегда простое использование ключевого слова `defer`. В этой главе также содержится информация о том, как использовать GitLab Runners и GitHub Actions для автоматизации, и о том, как создать образ Docker для двоичного файла Go.

Глава 6 посвящена системному программированию и содержит такие темы, как работа с аргументами командной строки, обработка сигналов UNIX, ввод и вывод файлов, интерфейсы `io.Reader` и `io.Writer` и использование пакетов `viper` и `cobra`. Кроме того, мы поговорим о работе с файлами JSON, XML и YAML, создадим удобную утилиту командной строки, позволяющую обнаруживать циклы в файловой системе UNIX, и обсудим встраивание файлов в двоичные файлы Go, а также функцию `os.ReadDir()`, тип `os.DirEntry` и пакет `io/fs`. Наконец, мы обновим приложение телефонной книги, чтобы можно было использовать данные JSON, и преобразуем его в соответствующую утилиту командной строки с помощью пакета `cobra`.

В главе 7 обсуждаются горутины, каналы и конвейеры. Мы рассмотрим различия между процессами, потоками и горутинами, пакет `sync` и то, как работает

планировщик Go. Кроме того, мы исследуем использование ключевого слова `select` и обсудим различные «типы» каналов Go, а также общую память, мьютексы, типы `sync.Mutex` и `sync.RWMutex`. В остальной части главы мы рассмотрим пакеты `context` и `semaphore`, рабочие пулы и выясним, как устанавливать тайм-аут для горутин и выявлять состояние гонки.

В главе 8 обсуждаются пакет `net/http`, разработка веб-серверов и веб-сервисов, предоставление метрик в Prometheus, визуализация метрик в Grafana, создание веб-клиентов и файловых серверов. Мы также преобразуем приложение телефонной книги в веб-сервис и создадим для него клиент командной строки.

Глава 9 посвящена пакету `net`, TCP/IP и протоколам TCP и UDP, а также сокетам UNIX и протоколу WebSocket. В этой главе мы разработаем множество сетевых серверов и клиентов.

В главе 10 описана работа с REST API и сервисами RESTful. Мы выясним, как определять REST API и разрабатывать мощные параллельные серверы RESTful, а также утилиты командной строки, которые действуют как клиенты сервисов RESTful. Наконец, мы познакомимся со Swagger, позволяющим создавать документацию для REST API, и выясним, как загружать двоичные файлы.

В главе 11 обсуждаются тестирование, оптимизация и профилирование кода, а также кросс-компиляция, сравнительный анализ кода Go, создание примеров функций, использование директивы `go:generate` и поиск недоступного кода Go.

Глава 12 посвящена работе с gRPC в Go. gRPC — это альтернатива сервисам RESTful, разработанная Google. В этой главе вы узнаете, как определить методы и сообщения сервиса gRPC, как перевести их в код Go и как разработать сервер и клиент для этого сервиса gRPC.

Глава 13 посвящена дженерикам и тому, как использовать новый синтаксис для написания универсальных функций и определения универсальных типов данных. Дженерики появились в версии Go 1.18, которая, согласно циклу разработки Go, была официально выпущена в марте 2022 года.

В приложении рассказывается о работе сборщика мусора Go и показано, как этот компонент Go может повлиять на производительность вашего кода.

Как сделать книгу максимально полезной

Для работы с книгой требуется компьютер UNIX с относительно недавно установленной версией Go, то есть это может быть любая машина, работающая под управлением macOS X, macOS или Linux. Бóльшая часть представленного

кода также будет выполняться на компьютерах с Microsoft Windows без каких-либо изменений.

Чтобы извлечь из книги максимальную пользу, попробуйте как можно скорее применить знания из каждой главы в собственных программах и посмотреть, что работает, а что нет! Как я уже говорил, попытайтесь выполнить упражнения, приведенные в конце каждой главы, или реализуйте свои программные задачи.

Файлы с примерами кода

Пакет кода для книги размещен на GitHub по адресу https://github.com/mactsouk/mastering-Go-3rd. У нас есть и другие пакеты кода из нашего богатого каталога книг и видео, доступные по адресу https://github.com/PacktPublishing/. Не забудьте в них заглянуть!

Цветные иллюстрации

Мы также предоставляем PDF-файл с цветными оригинальными снимками экрана и схемами из этой книги. Вы можете скачать его здесь: https://static.packt-cdn.com/downloads/9781801079310_ColorImages.pdf.

Условные обозначения

В книге используются следующие условные обозначения.

Кодовые слова в тексте, имена папок, имена файлов и их расширения, пути и пользовательский ввод оформляются моноширинным шрифтом. Например: «Звезда этой главы — пакет `net/http`, который содержит функции, позволяющие разрабатывать мощные веб-серверы и веб-клиенты».

Блок кода оформляется следующим образом:

```
package main

import (
    "fmt"
    "math/rand"
    "os"
    "path/filepath"
    "strconv"
    "time"
)
```

Когда мы хотим привлечь ваше внимание к определенной части блока кода, соответствующие строки или элементы выделяются **моноширинным жирным шрифтом**:

```
package main

import (
    "fmt"
    "math/rand"
    "os"
    "path/filepath"
    "strconv"
    "time"
)
```

Любой ввод или вывод из командной строки оформляется следующим образом:

```
$ go run www.go
Using default port number: :8001
Served: localhost:8001
```

Шрифтом без засечек оформляются URL или слова, которые вы видите на экране, например в меню или диалоговых окнах.

Новые термины и важные слова выделены *курсивом*.

Этот рисунок указывает на предупреждения или важные примечания.

Этот рисунок указывает на советы и рекомендации.

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства www.piter.com вы найдете подробную информацию о наших книгах.