



Full-stack web development with Go : build your web applications quickly using the Go programming language and Vue.js /

Tolaram, Nanik,
author

Monografia

A Go programmer's hands-on guide to go from building basic server-side templates to full single page apps using Tailwind CSS, Vue.js, and more Key Features Learn how to structure your Go projects, be smarter with SQLc, create middleware and secure your apps Explore different frontend frameworks and dive into Vue and Tailwind CSS Use Terraform and Docker and deploy in minutes Book Description Go is a modern programming language with capabilities to enable high-performance app development. With its growing web framework ecosystem, Go is a preferred choice for building complete web apps. This practical guide will enable you to take your Go skills to the next level building full stack apps. This book walks you through creating and developing a complete modern web service from auth, middleware, server-side rendering, databases, and modern frontend frameworks and Go-powered APIs. You'll start by structuring the app and important aspects such as networking, before integrating all the different parts together to build a complete web product. Next, you'll learn how to build and ship a complete product by starting with the fundamental building blocks of creating a Go backend. You'll apply best practices for cookies, APIs, and security, and level up your skills with the fastest growing frontend framework, Vue. Once your full stack application is ready, you'll understand how to push the app to production and be prepared to serve customers and share it with the world. By the end of this book, you'll have learned how to build and ship secure, scalable, and complete products and how to combine Golang with existing products using best practices. What you will learn Discover how to quickly scaffold, structure, and build your Golang applications Design and build databases using PostgreSQL and SQLc to generate type-safe SQL code Monitor applications with centralized logging and metrics collection for high performance Explore frontend frameworks and libraries such as Vuetify, Buefy, and Tailwind CSS to build a responsive frontend for your app Communicate securely between the frontend and backend with cookies and session handling with middleware Understand how to package applications using Docker and set up CI pipelines in GitHub Who this book is for This book is for Go programmers who want to combine the power of the Go language with modern web development technologies to build complete web apps from frontend to backend. The book assumes beginner-level knowledge of HTML, CSS, and the web

Título: Full-stack web development with Go build your web applications quickly using the Go programming language and Vue.js Nanik Tolaram, Nick Glynn

Edición: 1st ed

Editorial: Birmingham Mumbai Packt Publishing [2023] 2023

Descripción física: 1 online resource (302 pages)

Nota general: Includes index

Contenido: Cover -- Title Page -- Copyright and Credits -- Contributors -- About the reviewers -- Table of Contents -- Preface -- Part 1: Building a Golang Backend -- Chapter 1: Building the Database and Model -- Technical requirements -- Installing Docker -- Setting up Postgres -- Designing the database -- Installing sqlc -- Using sqlc -- Setting up the database -- Generating CRUD with sqlc -- Building the makefile -- Summary -- Chapter 2: Application Logging -- Technical requirements -- Exploring Go standard logging -- Using golog -- Local logging -- Writing log messages to the logging server -- Configuring multiple outputs -- Summary -- Chapter 3: Application Metrics and Tracing -- Technical requirements -- Understanding OpenTelemetry -- The OpenTelemetry APIs and SDK -- Tracing applications -- Installing Jaeger -- Integrating the Jaeger SDK -- Integration with Jaeger -- Adding metrics using Prometheus -- Adding metrics using Prometheus -- Running docker-compose -- Summary -- Part 2: Serving Web Content -- Chapter 4: Serving and Embedding HTML Content -- Technical requirements -- Handling HTTP functions and Gorilla Mux -- Hello, World with defaults -- Building on the basics with Gorilla Mux -- Rendering static content -- Rendering dynamic content -- Using Go embed to bundle your content -- Summary -- Chapter 5: Securing the Backend and Middleware -- Technical requirements -- Adding authentication -- Creating our dummy user -- Authenticating a user -- Adding middleware -- Basic middleware -- Adding cookies and sessions -- Cookies and session handling -- Storing session information -- Using Redis for a session -- Summary -- Chapter 6: Moving to API-First -- Technical requirements -- Structuring an application -- Defining packages -- Exposing our REST API -- Cross-Origin Resource Sharing (CORS) -- JSON middleware -- Session middleware Converting to and from JSON -- Defining request model -- Defining a response model -- Reporting errors with JSON -- Using JSONError -- Using JSONMessage -- Summary -- Part 3: Single-Page Apps with Vue and Go -- Chapter 7: Frontend Frameworks -- Technical requirements -- Server-side rendering versus single-page apps -- Introducing React, Vue, and more -- React -- Svelte -- Vue -- Creating a Vue app -- Application and components -- Login page using Vue -- Using Vite -- Using Vue Router to move around -- Routing the login page -- Summary -- Chapter 8: Frontend Libraries -- Technical requirements -- Understanding Vuetify -- Setting up Vuetify -- Using UI components -- Understanding Buefy -- Bulma sample -- Setting up Buefy -- UI components -- Validating data entry with Vuelidate -- Better input handling with Cleave.JS -- Summary -- Chapter 9: Tailwind, Middleware, and CORS -- Technical requirements -- Introducing Tailwind -- Creating a new Tailwind and Vite project -- Consuming your Golang APIs -- CORS for secure applications -- Creating Vue middleware -- Summary -- Chapter 10: Session Management -- Technical requirements -- Session management and JWTs -- What's a JWT? -- (Re)introducing Vue Router -- Navigation guards -- Defaults and error pages -- Summary -- Part 4: Release and Deployment -- Chapter 11: Feature Flags -- Technical requirements -- An introduction to feature flags -- Feature flag configuration -- Use cases for using feature flags -- Installing the feature flag server -- The high-level architecture of feature flags -- Integration of the feature flag -- Web application -- Microservice integration -- Summary -- Chapter 12: Building Continuous Integration -- Technical requirements -- Importance of CI -- Setting up GitHub -- GitHub Actions -- Publishing Docker images -- Dockerfile -- GitHub Packages -- Publishing to GitHub Packages Pulling from GitHub Packages -- Summary -- Chapter 13: Dockerizing an Application -- Technical requirements -- Installing Docker -- Using Docker -- Docker images -- Running images as containers -- Building and packaging images -- Docker Compose -- Summary -- Chapter 14: Cloud Deployment -- Technical requirements -- AWS refresher -- Amazon Elastic Compute Cloud -- Virtual Private Cloud -- Database storage -- Elastic Container Service -- AWS tools -- Understanding and using Terraform -- What is Terraform? -- Installing Terraform -- Terraform basic commands -- The init command -- The plan command -- The destroy command -- Coding in Terraform -- Providers -- Terraform examples -- Pulling from GitHub Packages -- AWS EC2 setup -- Deploying to ECS with a load balancer -- Summary -- Index -- Other Books You May Enjoy

ISBN: 9781803243917 electronic bk.)

Materia: Go (Computer program language) Web applications- Programming

Autores: Glynn, Nick, author

Enlace a formato físico adicional: Print version Tolaram, Nanik. Full-Stack Web Development with Go
Birmingham : Packt Publishing, Limited,c2023

Baratz Innovación Documental

- Gran Vía, 59 28013 Madrid
- (+34) 91 456 03 60
- informa@baratz.es