

The Lessons

A WHOLE LOT OF THEM!

📁 Section 1: Getting Started

- ⌚ A Basic Web Application ([Sample](#))
- ⌚ Troubleshooting and Slack
- ⌚ Packages and Imports ([Sample](#))
- ⌚ Editors and Automatic Imports ([Sample](#))
- ⌚ The "Hello, World" Part of our Code ([Sample](#))
- ⌚ Web Requests ([Sample](#))
- ⌚ HTTP Methods ([Sample](#))
- ⌚ Our Handler Function ([Sample](#))
- ⌚ Registering our Handler Function... ([Sample](#))
- ⌚ Go Modules ([Sample](#))

📁 Section 2: Adding New Pages

- ⌚ Dynamic Reloading ([Sample](#))
- ⌚ Setting Header Values ([Sample](#))
- ⌚ Creating a Contact Page ([Sample](#))
- ⌚ Examining the `http.Request` Type ([Sample](#))
- ⌚ Custom Routing ([Sample](#))
- ⌚ URL Path vs RawPath
- ⌚ Not Found Page
- ⌚ The `http.Handler` Type

- ⌚ The `http.HandlerFunc` Type
- ⌚ Exploring Handler Conversions
- ⌚ FAQ Exercises

📁 Section 3: Routers and 3rd Party Libraries

- ⌚ Defining our Routing Needs
- ⌚ Using git
- ⌚ Installing Chi
- ⌚ Using Chi
- ⌚ Chi Exercises

📁 Section 4: Templates

- ⌚ What are Templates?
- ⌚ Why Do We Use Server Side Rendering?
- ⌚ Creating Our First Template
- ⌚ Cross Site Scripting (XSS)
- ⌚ Alternative Template Libraries
- ⌚ Contextual Encoding
- ⌚ Home Page via Template
- ⌚ Contact Page via Template
- ⌚ FAQ Page via Template
- ⌚ Template Exercises

📁 Section 5: Code Organization

- ⌚ Code Organization
- ⌚ MVC Overview
- ⌚ Walking Through a Web Request with MVC
- ⌚ MVC Exercises

📁 Section 6: Starting to Apply MVC

- ⌚ Creating the Views Package
- ⌚ fmt.Errorf
- ⌚ Validating Templates at Startup
- ⌚ Must Functions
- ⌚ Exercises

📁 Section 7: Enhancing our Views

- ⌚ Embedding Template Files
- ⌚ Variadic Parameters
- ⌚ Named Templates
- ⌚ Dynamic FAQ Page
- ⌚ Reusable Layouts
- ⌚ Tailwind CSS
- ⌚ Utility-first CSS
- ⌚ Adding a Navigation Bar
- ⌚ Exercises

📁 Section 8: The Signup Page

- ⌚ Creating the Signup Page
- ⌚ Styling the Signup Page
- ⌚ Intro to REST
- ⌚ Users Controller
- ⌚ Decouple with Interfaces
- ⌚ Parsing the Signup Form
- ⌚ URL Query Params
- ⌚ Exercises

📁 Section 9: Databases and PostgreSQL

- ⌚ Intro to Databases

- ⌚ Intalling Postgres
- ⌚ Connecting to Postgres
- ⌚ Update: Docker Container Names
- ⌚ Creating SQL Tables
- ⌚ Postgres Data Types
- ⌚ Postgres Constraints
- ⌚ Creating a Users Table
- ⌚ Inserting Records
- ⌚ Querying Records
- ⌚ Filtering Queries
- ⌚ Updating Records
- ⌚ Deleting Records
- ⌚ Additional SQL Resources

📁 Section 10: Using Postgres with Go

- ⌚ Connecting to Postgres with Go
- ⌚ Imports with Side Effects
- ⌚ Postgres Config Type
- ⌚ Executing SQL with Go
- ⌚ Inserting Records with Go
- ⌚ SQL Injection
- ⌚ Acquire a new Record's ID
- ⌚ Querying a Single Record
- ⌚ Creating Sample Orders
- ⌚ Querying Multiple Records
- ⌚ ORMs vs SQL
- ⌚ Exercises
- ⌚ Syncing the Book and Screencasts Source Code

📁 Section 11: Securing Passwords

- ⌚ Steps for Securing Passwords
- ⌚ Third Party Authentication Options
- ⌚ What is a Hash Function?
- ⌚ Store Password Hashes, Not Encrypted or Plaintext Values
- ⌚ Salt Passwords
- ⌚ Learning bcrypt with a CLI
- ⌚ Hashing Passwords with bcrypt
- ⌚ Comparing a Password with a bcrypt Hash

📁 Section 12: Adding Users to our App

- ⌚ Defining the User Model
- ⌚ Creating the UserService
- ⌚ Create User Method
- ⌚ Postgres Config for the Models Package
- ⌚ UserService in the Users Controller
- ⌚ Create Users on Signup
- ⌚ Sign In View
- ⌚ Authenticate Users
- ⌚ Process Sign In Attempts

📁 Section 13: Remembering Users with Cookies

- ⌚ Stateless Servers
- ⌚ Creating Cookies
- ⌚ Viewing Cookies with Chrome
- ⌚ Viewing Cookies with Go
- ⌚ Securing Cookies from XSS
- ⌚ Cookie Theft
- ⌚ CSRF Attacks
- ⌚ CSRF Middleware

- ⌚ Providing CSRF to Templates via Data
- ⌚ Custom Template Functions
- ⌚ Adding the HTTP Request to Execute
- ⌚ Request Specific CSRF Template Function
- ⌚ Template Function Errors
- ⌚ Securing Cookies from Tampering
- ⌚ Cookie Exercises

📁 Section 14: Sessions

- ⌚ Random Strings with crypto/rand
- ⌚ Exploring math/rand
- ⌚ Wrapping the crypto/rand package
- ⌚ Why Do We Use 32 Bytes for Session Tokens?
- ⌚ Defining the Sessions Table
- ⌚ Stubbing the SessionService
- ⌚ Sessions in the Users Controller
- ⌚ Cookie Helper Functions
- ⌚ Create Session Tokens
- ⌚ Refactor the rand Package
- ⌚ Hash Session Tokens
- ⌚ Insert Sessions into the Database
- ⌚ Update Existing Sessions
- ⌚ Query Users via Session Token
- ⌚ Deleting Session
- ⌚ Sign Out Handler
- ⌚ Sign Out Link
- ⌚ Session Exercises

📁 Section 15: Improved SQL

- ⌚ SQL Relationships

- ⌚ Foreign Keys
- ⌚ On Delete Cascade
- ⌚ Inner Join
- ⌚ Left, Right, and Full Outer Join
- ⌚ Using Join in the SessionService
- ⌚ SQL Indexes
- ⌚ Creating PostgreSQL Indexes
- ⌚ On Conflict
- ⌚ Improved SQL Exercises

📁 Section 16: Schema Migrations

- ⌚ What are Schema Migrations?
- ⌚ How Migration Tools Work
- ⌚ Installing pressly/goose
- ⌚ Converting to Schema Migrations
- ⌚ Schema Versioning Problem
- ⌚ Run Goose with Go
- ⌚ Embedding Migrations
- ⌚ Go Migration Files
- ⌚ Removing Old SQL Files

📁 Section 17: Current User via Context

- ⌚ Using Context to Store Values
- ⌚ Improved Context Keys
- ⌚ Context Values with Types
- ⌚ Storing Users as Context Values
- ⌚ Reading Request Context Values
- ⌚ Set the User via Middleware
- ⌚ Requiring a User via Middleware

- ⌚ Accessing the Current User in Templates
- ⌚ Request-Scope Values

📁 Section 18: Sending Emails to Users

- ⌚ Password Reset Overview
- ⌚ SMTP Services
- ⌚ Building Emails with SMTP
- ⌚ Sending Emails with SMTP
- ⌚ Building an Email Service
- ⌚ EmailService.Send
- ⌚ Forgot Password Email
- ⌚ ENV Variables

📁 Section 19: Completing the Authentication System

- ⌚ Password Reset DB Migration
- ⌚ Password Reset Service Stubs
- ⌚ Forgot Password HTTP Handler
- ⌚ Asynchronous Emails
- ⌚ Forgot Password HTML Template
- ⌚ Initializing Services with ENV Vars
- ⌚ Check Your Email HTML Template
- ⌚ Reset Password HTTP Handlers
- ⌚ Reset Password HTML Template
- ⌚ Update Password Function
- ⌚ Implementing PasswordReset.Create
- ⌚ Implementing PasswordReset.Consume

📁 Section 20: Better Errors

- ⌚ Inspecting Errors
- ⌚ Inspecting Wrapped Errors

- ⌚ Designing the Alert Banner
- ⌚ Dynamic Alerts
- ⌚ Removing Alerts with JavaScript
- ⌚ Detecting Existing Emails
- ⌚ Accepting Errors in Templates
- ⌚ Public vs Internal Errors
- ⌚ Creating Public Errors
- ⌚ Using Public Errors
- ⌚ Better Error Handling Exercises

📁 Section 21: Galleries

- ⌚ Galleries Overview
- ⌚ Gallery Model and Migration
- ⌚ Creating Gallery Records
- ⌚ Querying for Galleries by ID
- ⌚ Querying Galleries by UserID
- ⌚ Updating Gallery Records
- ⌚ Deleting Gallery Records
- ⌚ New Gallery Handler
- ⌚ views.Template Name Bug
- ⌚ New Gallery Template
- ⌚ Gallery Routing and CSRF Bug Fixes
- ⌚ Create Gallery Handler
- ⌚ Edit Gallery Handler
- ⌚ Edit Gallery Template`
- ⌚ Update Gallery Handler
- ⌚ Gallery Index Handler
- ⌚ Discovering and Fixing a Gallery Index Bug
- ⌚ Gallery Index Template Continued

- ⌚ Show Gallery Handler
- ⌚ Show Gallery Template and a Tailwind Update
- ⌚ Extracting Common Gallery Code
- ⌚ Extra Gallery Checks with Functional Options
- ⌚ Delete Gallery Handler
- ⌚ Gallery Exercises

📁 Section 22: Images

- ⌚ Images Overview
- ⌚ Setting Up Test Images
- ⌚ Adding the ImagesDir to the GalleryService
- ⌚ Globbing Image Files
- ⌚ Adding Filename and GalleryID to the Image Type
- ⌚ Adding Images to the Show Gallery Page
- ⌚ Show Image Handler
- ⌚ Querying for a Single Image
- ⌚ URL Path Escaping Image Filenames
- ⌚ Adding Images to the Edit Gallery Page
- ⌚ Delete Image Form
- ⌚ Delete Image Service Func
- ⌚ Delete Image Handler
- ⌚ Checking for Filename Vulnerabilities
- ⌚ Upload Image Form
- ⌚ Image Upload Handler
- ⌚ Creating Images in the GalleryService
- ⌚ Detecting Content Type
- ⌚ Rendering Content Type Errors

📁 Section 23: Preparing for Production

- ⌚ Loading All Config via ENV
- ⌚ Docker Compose Overrides
- ⌚ Building Tailwind Locally
- ⌚ Tailwind Via Docker
- ⌚ Serving Static Assets
- ⌚ Making main Easier to Test
- ⌚ Running our Go Server via Docker
- ⌚ Multi-Stage Docker Builds
- ⌚ Tailwind Production Build
- ⌚ Caddy Server via Docker

📁 Section 24: Deploying

- ⌚ Creating a Digital Ocean Droplet
- ⌚ Setting up DNS
- ⌚ Installing Git on the Server
- ⌚ Setting Up a Bare Git Repo
- ⌚ Setting Up a Local Git Repo
- ⌚ Checking Out Our Code on the Server
- ⌚ Email Sending Server Setup
- ⌚ Production .env File
- ⌚ Install Docker in Prod
- ⌚ Production Caddyfile
- ⌚ Production Data Directories
- ⌚ Running Our App in Prod
- ⌚ Post-receive Deploy Updates
- ⌚ Deploy via Git
- ⌚ Logging Services

Bonus: OAuth

- ⌚ Intro to OAuth
- ⌚ OAuth Example Code
- ⌚ Dropbox App Setup
- ⌚ Offline OAuth Demo
- ⌚ OAuth Tokens
- ⌚ Online vs Offline Access Types
- ⌚ Redirect URIs
- ⌚ OAuth Connect HTTP Handler
- ⌚ Determine Redirect URI Host
- ⌚ OAuth Routes and Config Setup
- ⌚ OAuth Callback Handler
- ⌚ Testing OAuth with API Calls

Bonus: Dropbox Chooser

- ⌚ Dropbox Chooser Overview
- ⌚ Embedding the Chooser
- ⌚ Images via Dropbox Form
- ⌚ Chooser Success Function
- ⌚ Images via URL Handler
- ⌚ Downloading Images
- ⌚ Create Images Without Seek
- ⌚ Concurrent Downloads
- ⌚ Using errgroup
- ⌚ Page Specific JS