

# Fullstack with Go

Creating a full-stack quiz application in (mostly) Go

6 Nov 2024

# Hi, I'm Valentin 🖐️

software engineer at  
SSH Communications Security

- Certified vampire 🇧🇪
- Fullstack dev and design nerd
- Areas of focus: privacy, security, open-source, typography, Jojo's Bizarre Adventure
- <https://vanntile.com>

## What is my tech stack?








### Languages

Go  Python  JavaScript TypeScript C/C++ Haskell 

### Markup

JSON YAML HTML Markdown MDX




### Frameworks

Astro  NextJS  React  Vue  Angular  Tailwind CSS   
FastAPI  Express Django Flask GTK/gtkmm unified

### Testing

Robot  Cypress Jest  JUnit



### Tools

Deno  NodeJS  npm  Docker  Jenkins 

### Standards & paradigms

FOSS OpenAPI REST GraphQL semantic web functional programming

### Platforms

GitHub  GitLab  Bitbucket  Linode Mastodon 

### Design tools & knowledge

Inkscape Figma GIMP SVG Typography Layout Visual identities

### Other snappy stuff

git Linux neovim GNOME Obsidian Jupyter Prettier

# Go for the truth

*or how to underestimate a small web application*

# What do we want?

- quiz with real / fake questions
- submission: demographics form
- showing results to the user
- saving users' answers for later data analysis

The following statement was written based on the prompt: "I'm a iOS because..."

I am an iOS user because, at the time, it was the only choice for the best camera phone, especially with Huawei out of the picture. I dislike a lot about iPhones—the hotspot issues, battery life, limited customization (though it's improved a bit), but they do offer slightly better privacy. Plus, with everything locked down, you can't mess up your security settings! And, let's be honest, at least it's just Apple doing the spying—not the whole world.

Fake

Real

# What do we want?

- password-protected admin page
- uploading questions
- downloading answers
- minimal, fast database

Uploading questions will remove all existing questions. If you just want to add new questions, first download the existing questions and append new ones in the proper format. Or let a professional do it ;)

Download questions

Upload questions

Download answers

## Existing questions

| ID | Claims  | Fake | Question   |
|----|---------|------|--|
| 0  | Android | real | I am an Android user because I want to have control over the files on my device. I want to be able to organise the folders, to synchronize folders with my chosen cloud service, to know that deleted files stay deleted. I don't want to be locked into a single big company                        |
| 1  | Android | real | I am an Android user because Android devices have innovative features faster than Apple devices, quite often years ahead. Android phones are now having folding form factors, while iPhones cycle three design styles between generations. I prefer to have something cool, like a smart flip phone. |
| 2  | iOS     | fake | I am an iOS user because it is a stable phone with beautiful animations. Everything looks great and moves fast. I never have to wait for applications and never have problem air-dropping to my Macbook or using my ipods.   |
|    |         |      | I am an iOS user because Android is insecure and can be hacked much more easily than iPhones. I want to keep   |

# Technologies used

- Echo - web framework
- templ - template rendering in Go
- sqlite - one-file database
- tailwindcss - CSS utility framework
- air-verse/air - live reload Go apps in development
- vanilla JavaScript
  - Turso's libsql - distributed sqlite
  - nalgeon/sqllean - time operations in sqlite

# Setup

*or you thought I'd skip this*

# Environment

```
1  # .env.example
2  GO_ENV=development
3  ADDRESS=:1323
4  CERT_PATH=./server.crt
5  KEY_PATH=./server.key
6  DATABASE_PATH=file:./local.db
7  DATABASE_PASSWORD=secret
8  ADMIN_USERNAME=user
9  ADMIN_PASSWORD=secret
10 QUESTIONS_PER_QUIZ=10
```



# Development setup

```
# install air  
go install github.com/air-verse/air@latest
```

# It's codin' time



or 🎵 *there comes the fun* 🎵

# Simple echo server

```
// main.go
package main

import (
    "io"
    "net/http"
    "os"

    "github.com/labstack/echo/v4"
    "github.com/labstack/gommon/log"
    "github.com/rs/zerolog"
    "github.com/ziflex/lecho/v3"
)

func main() {
    // ...
}
```

# Middleware

```
import (  
    // ...  
    "github.com/labstack/echo/v4/middleware"  
    "golang.org/x/time/rate"  
)  
  
func main() {  
    // Middleware  
    if !development {  
        e.Pre(middleware.HTTPSRedirect())  
    }  
    e.Use(middleware.RequestID())  
    e.Use(lecho.Middleware(lecho.Config{Logger: logger, HandleError: true}))  
    e.Use(middleware.RateLimiter(middleware.NewRateLimiterMemoryStore(rate.Limit(10))))  
    e.Use(middleware.Secure())  
    e.Use(middleware.Timeout())  
    e.Use(middleware.Gzip())  
    e.Use(middleware.BodyLimit("2K"))  
    e.Use(middleware.Recover())  
}
```

# Configuration

```
type AppConfig struct {  
    development    bool  
    address        string  
    crtPath        string  
    keyPath        string  
    dbPath         string  
    dbPassword     string  
    adminUsername  string  
    adminPassword  string  
    questionsPerQuiz int  
}  
  
func NewAppConfig() (*AppConfig, error) {  
    development := os.Getenv("GO_ENV") == "development"  
    // ...  
}
```

# sqlite initialization

```
// go.mod  
replace github.com/mattn/go-sqlite3 => github.com/jgiannuzzi/go-sqlite3 v1.14.17-0.20230223050003-85a15a7254f2
```

# Router

```
type Handler struct {  
    db          *sql.DB  
    questionsPerQuiz int  
    questionCount int  
}  
  
func (h *Handler) GetQuiz(c echo.Context) error {}  
func (h *Handler) PostQuiz(c echo.Context) error {}  
func (h *Handler) GetAdminPage(c echo.Context) error {}  
func (h *Handler) GetAnswers(c echo.Context) error {}  
func (h *Handler) UploadQuestions(c echo.Context) error {}
```

# Data model

```
const (  
    Group1 string = "Android"  
    Group2 string = "iOS"  
)  
  
var GROUPS = []string{Group1, Group2}  
  
var COUNTRIES = []string{  
    "Afghanistan",  
    // ...  
}
```



# templ-ates

```
// web/views/quiz.templ
package views

import "github.com/vanttile/go-for-the-truth/web/models"

templ Introduction(questionCount int) {
    <div id="pageIntro" class="flex min-h-screen items-center justify-center py-16">
        <div class="w-full max-w-prose">
            <h1>Go for the truth</h1>
            <p>
                Can you tell the difference between real and fake
                <span class="border-b-4 border-green-700">{ models.Group1 }</span> and
                <span class="border-b-4 border-slate-700">{ models.Group2 }</span> users?
            </p>
            <p>
                You're about to see { fmt.Sprintf("%d", questionCount) } statements that people wrote based one of two
                prompts:
            <!-- ... -->
        </div>
    </div>
}
```

# Layout and template composition

```
// web/views/layout.templ
templ BaseLayout(content templ.Component) {
  <!DOCTYPE html>
  <html lang="en">
    <head>
      <meta charset="utf-8"/>
      <link rel="icon" type="image/svg+xml" href="/favicon.svg"/>
      <meta name="viewport" content="width=device-width"/>
      <meta name="robots" content="index, follow"/>
      <meta name="author" content="vanntile"/>
      <meta property="date" content="2024-11-06T08:00:00Z"/>
      <title>Go for the truth</title>
      <link rel="stylesheet" type="text/css" href="/styles.css"/>
    </head>
    <body class="bg-slate-200 ease-in-out flex items-center justify-center min-h-screen">
      @content
    </body>
  </html>
}
```

# Styles in the wind

```
/* web/input.css */  
@tailwind base;  
@tailwind components;  
@tailwind utilities;
```

# Submission form

- get country, default to US
- choose a side
- age validation (18 - 120)
- input validation, dirty state, error message

Before we show you your results, we have a few more questions for you that are for informational purposes only.

Do you live in the United States? \*

No

In which country do you live? \*

Finland

If you had to choose one side to identify with, which would you choose? \*

Android

What is your age? \*

25

By submitting, you agree to the processing of the data above for further research means. See our [Privacy Policy](#) for more info

Send answers and see your results

# Client-side interactivity

```
// web/public/quiz.js
const togglePageVisible = (page, toVisible) => {
  if (toVisible) {
    page.classList.remove("hidden");
    page.classList.add("flex");
  } else {
    page.classList.add("hidden");
    page.classList.remove("flex");
  }
};

window.addEventListener("DOMContentLoaded", (_event) => {
  // ...

  // Form
  const countryUSEl = document.querySelector("#countryUS");
  const formCountryEl = document.querySelector("#formCountry");
  const countrySelectEl = document.querySelector("#countrySelect");
  const sideSelectEl = document.querySelector("#sideSelect");
  const ageEl = document.querySelector("#age");
  const errCls = ["border-red-700", "text-red-700", "ring-red-700"];
});
```

# Downloading answers

```
// web/public/admin.js
window.addEventListener("DOMContentLoaded", (_event) => {
  let downloadingAnswers = false;
  document.querySelector("#downloadAnswers").addEventListener("click", async () => {
    if (downloadingAnswers) return;
    downloadingAnswers = true;

    let answers = "created,seed,country,side,age,answeredFake,answeredReal\n";

    let created;
    let callCount = 60; // 60 * 4e3 = 240e3
    while (callCount > 0) {
      callCount--;

      const res = await fetch(`/admin/answers${created ? "?created=" + encodeURIComponent(created) : ""}`);
      if (res.status !== 200) {
        console.error(`failed to get answers (status = ${res.status})`);
      }
      // ...
    }
  });
});
```

# Reading answers by offset

```
func (h *Handler) GetAnswers(c echo.Context) error {
    var offset *time.Time
    created := c.QueryParam("created")
    if created != "" {
        if t, err := time.Parse(time.RFC3339Nano, created); err == nil {
            offset = &t
        }
    }
    answers, err := GetAnswers(h.db, 4000, offset)
    if err != nil {
        c.Logger().Errorf("failed to read answers: %v", err)
        return Render(c, http.StatusAccepted, views.BaseLayout(views.ErrorPage("Failed to read answers")))
    }

    answerStr := ""
    for i := range answers {
        answerStr += fmt.Sprintf("%s,%s,\"%s\\\",%s,%s,\"%s\\\", \"%s\\\"\\n\", answers[i].Created.Format(time.RFC3339Nano),
            answers[i].Seed, answers[i].Country, answers[i].Side, answers[i].Age, answers[i].Fake, answers[i].Real)
    }

    return c.String(http.StatusOK, answerStr)
}
```

**END**

Thank you for listening