

Basic Go HTTP Server - Project Documentation

Overview

This project is a simple HTTP server built in Go. It demonstrates routing, basic function handlers, and serving static HTML pages.

File Structure

go-server/

- index.html

- form.html

- main.go

Architecture (Explained)

The server handles three main routes:

1. '/' - Serves index.html (homepage)
2. '/hello' - Runs helloHandler, returns a plain text message
3. '/form' - Runs formHandler, serves form.html

Each route is registered using `http.HandleFunc` in `main.go`.

Routes and Functionalities

/ => Serves index.html

/hello => Executes helloHandler (Responds with plain text)

/form => Executes formHandler and serves form.html

Route Details

/:

- Purpose: Home route
- Function: Serves index.html
- Use Case: Main landing page

Basic Go HTTP Server - Project Documentation

/hello:

- Purpose: Function route
- Function: Returns plain text like 'Hello, world'
- Use Case: Demonstrates basic dynamic response

/form:

- Purpose: Handles form route
- Function: Processes logic, returns form.html
- Use Case: Used to simulate form interactions

Technologies Used

- Language: Go (Golang)
- Packages: net/http, html/template (if templating is used)
- Concepts:
 - * HTTP routing
 - * Static file serving
 - * Custom function handlers

How to Run the Server

1. In terminal, run:

```
go run main.go
```

2. Open browser and visit:

- `http://localhost:8080/` => index.html
- `http://localhost:8080/hello` => helloHandler response
- `http://localhost:8080/form` => form.html

Key Revision Notes

Basic Go HTTP Server - Project Documentation

- Use `http.HandleFunc()` to bind routes to functions
- Use `http.ServeFile()` to return static HTML
- Structure allows easy extension with more endpoints
- Great foundation for learning Go web servers