Learn the Go Programming Language

For experienced developers or those of an adventurous nature

gotutorial.net
@GoTutorialNet

Matt Nunogawa
@amattn

Lesson 08

Making a Testable Web App

`v0.1 draft`

WE'RE ONLY* GOING TO USE THE STANDARD LIBRARY

```
net/http
net/http/httptest
html/template
database/sql

bytes
fmt
io/ioutil
log
strconv
strings
time

github.com/lib/pq/
```

* PostgreSQL driver, not the standard library, but we don't call it directly.

How To Follow Along

```
# This repo is pdfs as well as code, so it’s a bit
large.

git clone https://github.com/amattn/gotutorial.git
cd gotutorial

git checkout tags/b01
git checkout tags/b02
git checkout tags/b03
...

# after any checkout you can:
go test
go build && ./gtls
```

BUILD 01

- This is just hello world.

NET/HTTP

ROUTERS & HANDLERS

- At a basic level, we use net/http to route requests to handlers
- The simplest possible way to do is with the built-in pattern matching and HandleFunc

BUILD 02

- HTTP Hello World
- Route using http.HandleFunc()
- Simplest Posible Pattern Matching & One handler
  - “/“
  - inline function literal

Build 03

- Some refactoring
- Route using http.Handle() & a dedicated handle struct
- Add some primitive logging

BUILD 04

- More refactoring of routing
- Route using http.ListenAndServe & a dedicated routing struct
- router calls handlers
- think about centralizing logging

Lots of 3rd Party routers

- gorilla/mux
- https://github.com/bmizerany/pat
- collectivehealth/eprouter

BUILD 05

- More refactoring of routing
- Simplify child handlers with an interface
- centralize logging in router

BUILD 06

- Update our router
  - route to Admin or LinksHandler as appropriate
  - modify our interface to support custom response headers
- Make an ultra-primitive "In-Memory DB" to store our short links
- Make a LinksHandler to do handle shortlink urls

NET/HTTP/HTTPTEST

BUILD 07

- Unit Testing!

```
httptest.NewServer(router)
reflect.DeepEqual(expected, candidate)
```

- Currently just checking response status code

BUILD 08

- New Abstract BaseHandler
- prototype an ugly POST route/handler

BUILD 09

- Working Post route

- Add a shortlink

```
curl --data "code=123&url=http://google.com" http://
localhost:8080/admin/post
```

- Test a shortlink

```
curl -I http://localhost:8080/123
```

HTML/TEMPLATE

Basic Template Usage

- Make
- Parse (“compile”)
- Execute

BUILD 10

- Refactor Router to cleanup admin handler
- Refactor admin handler to use html/template after adding a new shortlink

DATABASE/SQL

How to use database/sql

- Find a driver
- connect
- prepare
- exec or query

Build 11

- Use a real database

FURTHER READING

- http://go-database-sql.org
- https://code.google.com/p/go-wiki/wiki/SQLInterface
- http://golang.org/pkg/database/sql/
- http://gophercon.sourcegraph.com/post/83852708856/building-database-applications-with-database-sql

BUILD 12

- Slightly more complicated html/template example
- List all shortlinks

`http://localhost:8080/admin/list`

LARGER FRAMEWORKS

- revel
- martini
- beego

MORE READING

- http://golang.org/doc/articles/wiki/
- http://codegangsta.gitbooks.io/building-web-apps-with-go/

Thank You, Credits & License

http://gotutorial.net
@GoTutorialNet

- I owe many many, thanks to the many authors of Go and to Rob Pike in particular.
- These slides are Copyright 2013-2014 Matthew Nunogawa

Matt Nunogawa
@amattn