LESSON 12



Go

Build a Web API

Learn it all @MyWebUniversity



A FREE MYWEBUNIVERSITY COURSE

GO PROGRAMMING

LESSON 12

Project: A JSON Web API

net/http, routing, JSON encoding, a CLI flag, Docker and next steps

An HTTP Server in Ten Lines

main.go

```
func main() {
    mux := http.NewServeMux()
    mux.HandleFunc("GET /books", listBooks)
    mux.HandleFunc("GET /books/{id}", getBook)
    mux.HandleFunc("POST /books", addBook)
    log.Println("listening on :8080")
    log.Fatal(http.ListenAndServe(":8080", mux))
}
```

- net slash http is production ready. Register a handler for a route, then ListenAndServe. …
- Since Go 1.22 the router understands methods and path parameters, like GET slash books …

01

JSON with Struct Tags

books.go

```
type Book struct {
    ID     int    `json:"id"`
    Title  string `json:"title"`
    Author string `json:"author"`
}
func listBooks(w http.ResponseWriter, r *http.Request) {
    w.Header().Set("Content-Type", "application/json")
    json.NewEncoder(w).Encode(store.All())
}
```

- Struct tags in back quotes tell encoding slash json the field names to use. json dot …
- NewDecoder reads a JSON request body into a struct, returning an error for bad input.

02

Handling Input and Errors

books.go

```
func getBook(w http.ResponseWriter, r *http.Request) {
    id, err := strconv.Atoi(r.PathValue("id"))
    if err != nil {
        http.Error(w, "bad id", http.StatusBadRequest); return
    }
    b, ok := store.Get(id)
    if !ok { http.NotFound(w, r); return }
    json.NewEncoder(w).Encode(b)
}
```

- r dot PathValue reads a path parameter. Validate everything, and answer with the right …
- Protect the shared store with a mutex, because many requests run at the same time.

03

Try It and Ship It

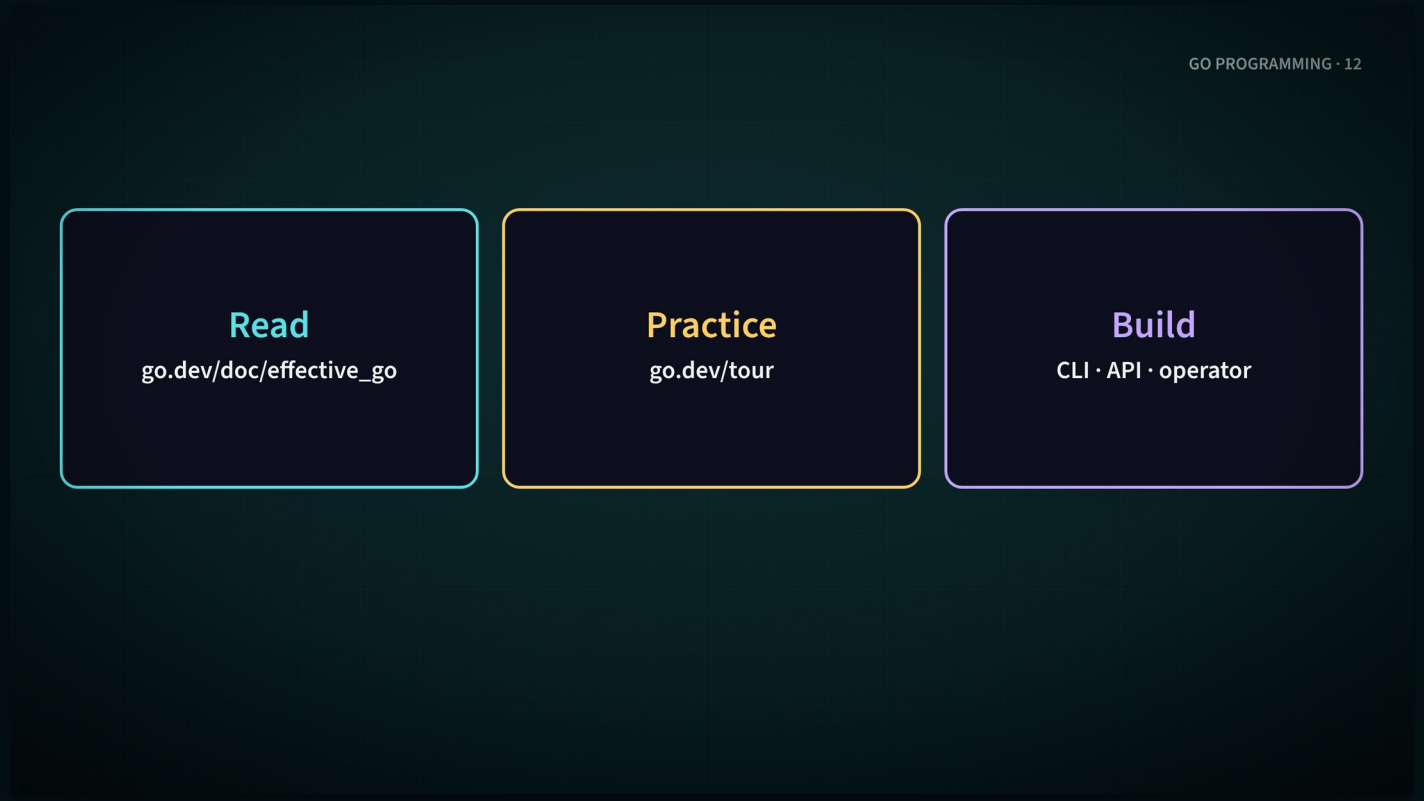
terminal

```
$ curl -X POST localhost:8080/books -d '{"title":"Shahnameh","author":"Ferdowsi"}'
{"id":1,"title":"Shahnameh","author":"Ferdowsi"}
$ curl localhost:8080/books/1
$ CGO_ENABLED=0 go build -o api . && docker build -t books-api .
```

- Test the API with curl. Then build a tiny Docker image: compile a static binary, and copy …
- The result is often under twenty megabytes and starts in milliseconds.

04

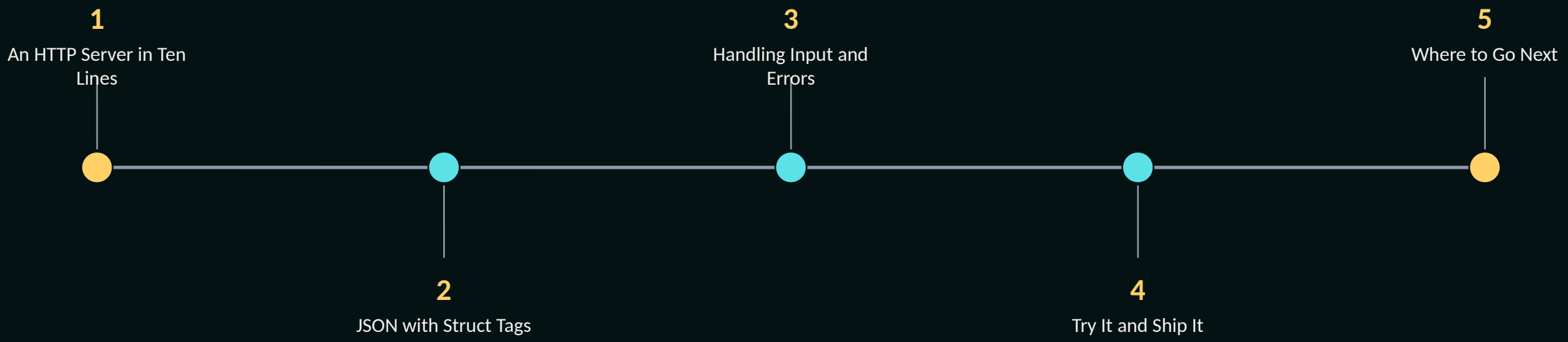
Where to Go Next



- Read — go.dev/doc/effective_go
- Practice — go.dev/tour
- Build — CLI · API · operator

05

Lesson Map


1
An HTTP Server in Ten Lines
2
JSON with Struct Tags
3
Handling Input and Errors
4
Try It and Ship It
5
Where to Go Next

Sources and Further Reading

main.go

```go
func main() {
    mux := http.NewServeMux()
    mux.HandleFunc("GET /books", listBooks)
    mux.HandleFunc("GET /books/{id}", getBook)
    mux.HandleFunc("POST /books", addBook)
    log.Println("listening on :8080")
    log.Fatal(http.ListenAndServe(":8080", mux))
}
```

GO PROGRAMMING · 12

books.go

```go
func getBook(w http.ResponseWriter, r *http.Request) {
    id, err := strconv.Atoi(r.PathValue("id"))
    if err != nil {
        http.Error(w, "bad id", http.StatusBadRequest); return
    }
    b, ok := store.Get(id)
    if !ok { http.NotFound(w, r); return }
    json.NewEncoder(w).Encode(b)
}
```

Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari