درس ۱۲

Go

ساختن web API

main.go

```
mux := http.NewServeMux()
mux.HandleFunc("GET /books", list)
http.ListenAndServe(":8080", mux)
```

همه را یاد بگیرید @MyWebUniversity-Dari

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی Go

درس ۱۲

پروژه: یک web API با JSON



net/http، routing، رمزگذاری JSON، یک بیرق CLI، Docker و قدم‌های بعدی

یک HTTP server در ده سطر

GO PROGRAMMING · 12

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))
}
```

net/http آمادهٔ تولید است. یک handler را برای یک route ثبت کنید، بعد ListenAndServe. هر ...

از Go 1.22 router متدها و پارامترهای مسیر را می‌فهمد، مثل GET /books/{id}.

01

JSON با struct tag ها

struct tag ها داخل back quote به encoding/json نام فیلدها را می‌گویند.
json.NewEncoder یک ...

NewDecoder یک درخواست را به JSON بدنهٔ یک struct می‌خواند، و برای ورودی بد یک error ...

books.go

```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())
}
```

02

مدیریت ورودی و خطاها

r.PathValue یک پارامتر مسیر را می‌خواند. همه چیز را بررسی کنید، و با status code درست پاسخ ...

store شریک را با یک mutex محافظت کنید، چون درخواست‌های زیاد همزمان اجرا می‌شوند.

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)
}
```

03

امتحان و انتشار

API را با curl امتحان کنید. بعد یک Docker image کوچک بسازید: یک فایل اجرایی static کمپایل ...

نتیجه اغلب کمتر از بیست مگابایت است و در چند میلی‌ثانیه شروع می‌شود.

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 .
```

04

قدم‌های بعدی

بخوانید — go.dev/doc/effective_go

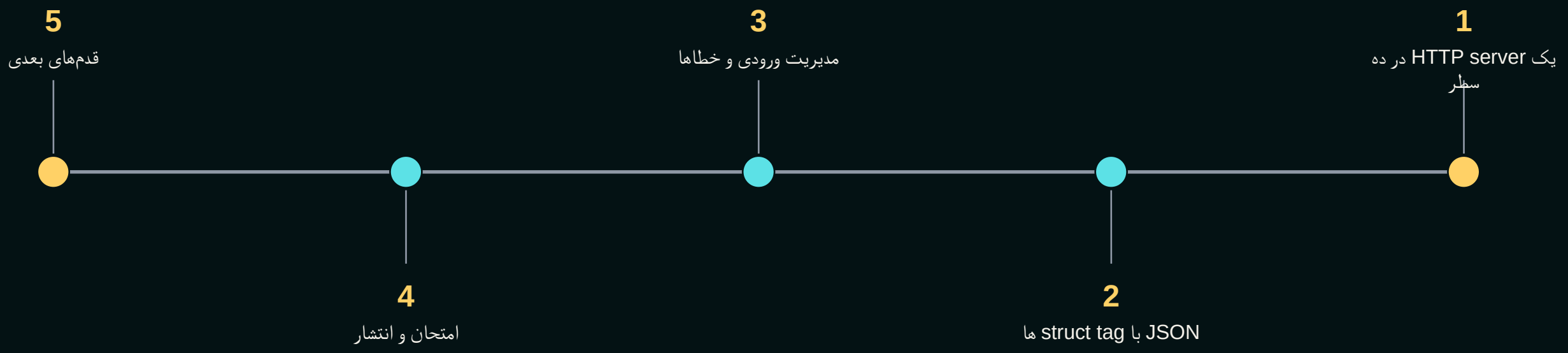
تمرین — go.dev/tour

بسازید — CLI · API · operator

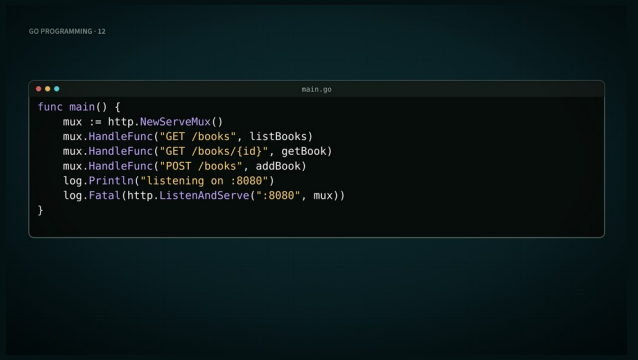


05

نقشهٔ درس


1
یک HTTP server در ده سطر
2
JSON با struct tag ها
3
مدیریت ورودی و خطاها
4
امتحان و انتشار
5
قدم‌های بعدی

منابع و مطالعهٔ بیشتر





تشکر از تماشای شما

MyWebUniversity.com

آموزش رایگان تکنالوژی از سال ۲۰۰۰ — انگلیسی و دری

@MyWebUniversity-Dari · @MyWebUniversity