

Литература

errgroup

- github.com/hashicorp/go-multierror#Group
- golang.org/x/sync/errgroup

В книгах

- **Concurrency in Go: Tools and Techniques for Developers** (Katherine Cox-Buday)
 - Concurrency at Scale: Error Propagation
- **Go Standard Library Cookbook** (Radomír Sohlich)
 - Fun with Concurrency: Propagating errors with errgroup

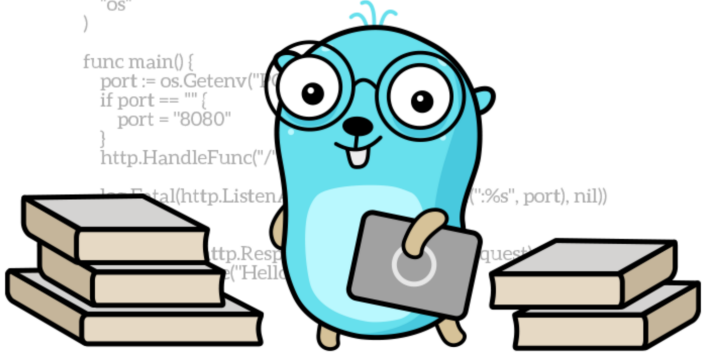
В сети

- [Atatus Blog: Justin: Goroutines Error Handling](#)
- [FullStory, Inc: Scott Blum: Why you should be using errgroup.WithContext\(\) in your Golang server handlers](#)
- [Golang Code: Edd Turtle: How to: Handle Errors within Wait Groups](#)
- [Medium: A Journey With Go: Vincent Blanchon: Go: Multiple Errors Management](#)
- [Medium: Statuscode: Claudio Fahey: Pipeline Patterns in Go. Pipeline with Error-Handling and Cancellation](#)
- [The Go Blog: Sameer Ajmani: Go concurrency Patterns: Pipelines and cancellation](#)
- [Хабр: Авито: Flaker: Как не ошибиться с конкурентностью в Go](#)

```
// This server can run on App Engine.
package main

import (
    "fmt"
    "log"
    "net/http"
    "os"
)

func main() {
    port := os.Getenv("PORT")
    if port == "" {
        port = "8080"
    }
    http.HandleFunc("/hello", func(w http.ResponseWriter, r *http.Request) {
        fmt.Fprintln(w, "Hello, world!")
    })
    log.Fatal(http.ListenAndServe(":"+port, nil))
}
```



go learn()