

# Литература

## Мнение

- [golang/go: Wiki: ExperienceReports: Error Values](#)
- [golang/go: Wiki: ExperienceReports: Error Handling](#)

## Go 2 Error Draft Designs

- [golang/go: Wiki: Go 2 Status](#)
- [golang/go: Issues: proposal: Go 2 error values](#)
- [Go 2 Draft Designs](#)
  
- [Go 2 Draft Designs: Error Values – Problem Overview \(Russ Cox\)](#)
- [Go 2 Draft Designs: Error Values Inspection – Draft Design \(Jonathan Amsterdam\)](#)
- [Go 2 Draft Designs: Error Values Printing – Draft Design \(Marcel van Lohuizen\)](#)
- [Go 2 Draft Designs: Error Values – Feedback](#)
  
- [Go 2 Draft Designs: Error Handling – Problem Overview \(Russ Cox\)](#)
- [Go 2 Draft Designs: Error Handling – Draft Design \(Marcel van Lohuizen\)](#)
- [Go 2 Draft Designs: Error Handling – Feedback](#)

## Error Handling Proposals

- [golang/go: Proposals: "error-handling" tag](#)
- [Robert Griesemer: Proposal: A built-in Go error check function, try](#)

## Прочее

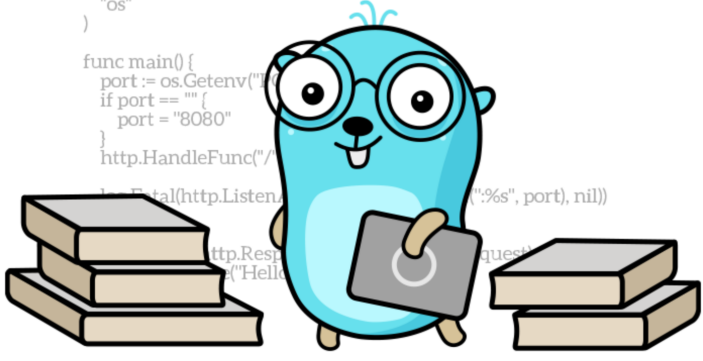
- [Brandur Leach: Error stack traces in Go with x/xerror](#)
- [Dave Cheney: Why Go gets exceptions right](#)
- [Dave Cheney: THE PAST, PRESENT, AND FUTURE OF GO 2](#)
- [Go: A Documentary \(Changkun Ou\): Error values \(1.13\)](#)
- [Frequently Asked Questions \(FAQ\): Why does Go not have exceptions?](#)
- [github.com/lainio/err2](#)
- [HACKERNOON: Harri Lainio: Golang – More Error Handling with Less Checking](#)
- [Microsoft Developer Blogs: Raymond Chen: Cleaner, more elegant, and wrong](#)
- [Microsoft Developer Blogs: Raymond Chen: Cleaner, more elegant, and harder to recognize](#)
- [Raphael Poss \(dr knz\): Errors vs. exceptions in Go and C++ in 2020](#)
- [Raul Jordan: Why Go's Error Handling is Awesome](#)
- [Reddit: r/golang: u/swyx: why no try/catch in golang? what's the theory behind this?](#)
- [The Go Blog: Russ Cox: Experiment, Simplify, Ship](#)
- [The Go Blog: Russ Cox: Toward Go 2](#)
- [The Go Blog: Go 2 Draft Designs](#)
- [YouTube: Gopher Academy: GopherCon 2017: Russ Cox - The Future of Go](#)
- [Хабр: divan0: Обработка ошибок в Go 2](#)

---

```
// 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()