

# Литература

## Истоки

- [github.com/golang/xerrors](https://github.com/golang/xerrors)

## github.com/pkg/errors

- [github.com/pkg/errors](https://github.com/pkg/errors)
- [golang/go: Issues: errors: ensure the 1.13 method to access the cause is called Cause\(\) not Unwrap\(\)](https://github.com/golang/go/issues/20134)
- [Dave Cheney: Stack traces and the errors package](https://dave.cheney.net/2016/04/27/stack-traces-and-the-errors-package)
- [Dmitry Harnitski: Can new Go errors wrapper replace pkg/errors?](https://dmitrybaranovskiy.github.io/issue-primer/#pull-request)
- [Stack Overflow: leventov: What's the difference between errors.Wrap\(\), errors.Errorf\(\), and fmt.Errorf\(\)?](https://stackoverflow.com/questions/48147480/what-is-the-difference-between-errors-wrap-errors-errorf-and-fmt-errorf)

## CocroachDB

- [github.com/cockroachdb/errors](https://github.com/cockroachdb/errors)
- [Cocroach: RFC 20190318: Error Handling](https://cockroachdb.com/docs/2019.1/rfcs/20190318-error-handling.html)

## Хорошая серия статей от Raphael Poss (aka dr knz, CocroachDB)

- [Part 1: The Go standard error APIs](https://drknz.org/2019/01/22/go-standard-error-apis/)
- [Part 2: Go's formatting APIs](https://drknz.org/2019/01/22/go-formatting-apis/)
- [Part 3: The Go error printing catastrophe](https://drknz.org/2019/01/22/go-error-printing-catastrophe/)
- [Part 4: Beyond fmt.Errorf\(\)—everyday error objects in Go](https://drknz.org/2019/01/22/beyond-fmt-errorf-everyday-error-objects-in-go/)

## Upspin

- [upspin.io/errors](https://upspin.io/errors)
- [Command Center: Rob Pike and Andrew Gerrand: Error handling in Upspin](#)
- [YouTube: The Go Programming Language: Gopherfest 2017: Upspin \(Rob Pike\)](#)

## emperror

- [github.com/emperror](https://github.com/emperror)
- [github.com/emperror/emperror](https://github.com/emperror/emperror)
- [github.com/emperror/errors](https://github.com/emperror/errors)

## Stack Trace

- [github.com/go-faster/errors](https://github.com/go-faster/errors)
- [github.com/palantir/stacktrace](https://github.com/palantir/stacktrace)
- [github.com/snwfdhmp/errlog](https://github.com/snwfdhmp/errlog)
- [github.com/ztrue/tracerr](https://github.com/ztrue/tracerr)
- [Brandur Leach: Error stack traces in Go with x/xerror](#)
- [DEV Community: Steve Coffman: Comparison golang stacktrace error library output](#)
- [Komu Wairagu: Build a Go package that annotates errors with stack traces](#)

## Не очень популярные

- [github.com/bnkamalesh/errors](https://github.com/bnkamalesh/errors)
- [github.com/ddymko/go-jsonerror](https://github.com/ddymko/go-jsonerror)
- [github.com/go-errgo/errgo](https://github.com/go-errgo/errgo)
- [github.com/go-errors/errors](https://github.com/go-errors/errors)
- [github.com/hashicorp/errwrap](https://github.com/hashicorp/errwrap)
- [github.com/joomcode/errorx](https://github.com/joomcode/errorx)

- [github.com/juju/errgo](https://github.com/juju/errgo)
- [github.com/neuronlabs/errors](https://github.com/neuronlabs/errors)
- [github.com/PumpkinSeed/errors](https://github.com/PumpkinSeed/errors)
- [github.com/rotisserie/eris](https://github.com/rotisserie/eris)
- [github.com/rozturac/cerror](https://github.com/rozturac/cerror)
- [github.com/segmentio/errors-go](https://github.com/segmentio/errors-go)
- [github.com/SonicRoshan/falcon](https://github.com/SonicRoshan/falcon)
- [github.com/spacemonkeygo/errors](https://github.com/spacemonkeygo/errors)
- [github.com/src-d/go-errors](https://github.com/src-d/go-errors)
- [github.com/vtopc/verrors](https://github.com/vtopc/verrors)

## Полезные

- [github.com/uber-go/multierr](https://github.com/uber-go/multierr)
- [github.com/hashicorp/go-multierror](https://github.com/hashicorp/go-multierror)
- [github.com/g8rswimmer/error-chain](https://github.com/g8rswimmer/error-chain)

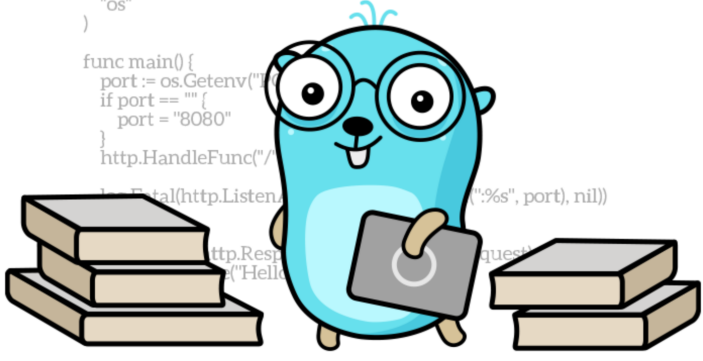
---

Делитесь своими ссылками в комментариях :)

```
// 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()