

Литература

errors

- [A Tour of Go: Errors](#)
- [Frequently Asked Questions \(FAQ\): Why is my nil error value not equal to nil?](#)
- [golang/go: Wiki: Errors](#)
- [Go talks: Rob Pike: Go at Google: Language Design in the Service of Software Engineering: 16. Errors](#)
- [Golang: Package errors](#)
- [Golang Tutorial: Return and handle an error](#)
- [The Go Blog: Andrew Gerrand: Error handling and Go](#)
- [The Go Blog: Rob Pike: Errors are values](#)
- [The Go Blog: Russ Cox: Experiment, Simplify, Ship: Errors](#)
- [Wiki: Go - Обработка ошибок и исключительных ситуаций](#)

Dave Cheney

- [Dave Cheney: Constant errors](#)
- [Don't force allocations on the callers of your API](#)
- [Don't just check errors, handle them gracefully](#)
- [Eliminate error handling by eliminating errors](#)
- [Inspecting errors](#)
- [Let's talk about logging](#)
- [Practical Go: Real world advice for writing maintainable Go program: Error Handling](#)
- [YouTube: dotconferences: dotGo 2019 - Dave Cheney - Constant Time](#)

Стиль

- [golang/go: Wiki: CodeReviewComments: Error Strings](#)
- [golang/go: Wiki: CodeReviewComments: Handle errors](#)
- [golang/go: Wiki: CodeReviewComments: In-Band Errors](#)
- [golang/go: Wiki: CodeReviewComments: Indent Error Flow](#)
- [golang/go: Wiki: CodeReviewComments: Receiver Type](#)
- [Uber Go Style Guide: Error Types](#)
- [Uber Go Style Guide: Error Wrapping](#)
- [Uber Go Style Guide: Linting](#)

YouTube

- [dotconferences: dotGo 2017 - John Cinnamon - Go Lift](#)
- [dotconferences: dotGo 2019 - Dave Cheney - Constant Time](#)
- [Gopher Academy: GopherCon 2016: Dave Cheney - Dont Just Check Errors Handle Them Gracefully](#)
- [Gopher Academy: GopherCon 2020: Jonathan Amsterdam - Working with Errors](#)
- [Golang Channel: Еще раз про обработку ошибок / Никита Галушко \(Mail.Ru Group\)](#)
- [SberMarket Tech: Дмитрий Лукиянчук: Как взять контроль над ошибками в свои руки и начать жить](#)
- [Web3Coach: How to configure an `if err != nil` macro in GoLand](#)

В книгах

- **Learning Functional Programming in Go** (Lex Sheehan)
- **Go101** (Tapir Liu)
 - Two error values returned by two errors. New calls with the same argument are

not equal.

- We should use `os.IsNotExist(err)` instead of `err == os.ErrNotExist` to check whether or not a file exists.

и т.д.

- **The Go Bestiary: Bugs, gotchas and guidelines for go programmers** (Kenneth Grant)

- Always handle Errors.

- Don't compare interface to nil.

- Either log or return an error.

- Error strings.

- Errors & Logging.

- Stack traces on errors.

и т.д.

Константные ошибки

- [Medium: Travix Engineering: Ralph Liqtenberg: Errors derived from constants in Go](#)
- [Reddit: r/golang: u/dgryski: Constant errors](#)
- [uber-go/guide: Issues: use const for custom error type](#)
- [YouTube: dotconferences: dotGo 2019 - Dave Cheney - Constant Time](#)

Логирование ошибок

- [Banzai Cloud: Márk Sági-Kazár: Error handling practices in Go: Extracting context from an error](#)
- [Dave Cheney: Let's talk about logging](#)
- [sumo logic: DevOps and Security Glossary Terms: Structured Logging](#)
- [uber-go/zap: Issues: Errors with attached structured data](#)

Monadic Errors

- [Dave Cheyney: Eliminate error handling by eliminating errors](#)
- [David Golden: Schrödinger's Monad: A Maybe type experiment for Go](#)
- [INNOQ: Martin Kühl: Error Handling in Go: Rob Pike Reinvented Monads](#)
- github.com/asteris-llc/gofpher
- github.com/onemouth/go-errormonad
- github.com/rbrahul/gofp
- github.com/seborama/fuego
- github.com/TeaEntityLab/fpGo
- github.com/tobyhede/go-underscore
- github.com/xdg-go/maybe
- Learning Functional Programming in Go (Lex Sheehan)
- [Rebecca Skinner: Monadic Error Handling in Go](#)
- [The Go Blog: Rob Pike: Errors are values](#)
- [Walter Schulze: Monads for Go Programmers](#)
- [YouTube: dotconferences: dotGo 2017 - John Cinnamon - Go Lift](#)
- [Хабр: Mail.ru Group: AloneCoder: Монады для Go-программистов](#)

Railway Oriented Programming

- [F# for Fun and Profit: Scott Wlaschin: Railway Oriented Programming](#)
- [Хабр: marshinov: Железнодорожно-ориентированное программирование. Обработка ошибок в функциональном стиле](#)

Статический анализ работы с ошибками

- [depguard](#)
- [errcheck](#)

- [errname](#)
- [errors](#)
- [errwrap](#)
- [forbidigo](#)
- [go-err113](#)
- [go-errorlint](#)
- [nilerr](#)
- [rowserrcheck](#)
- [ruleguard](#)
- [wrapcheck](#)

Прочее

- [8th Light: Kyle Krull: Exploring Error Handling Patterns in Go](#)
- [Ardan Labs: William Kennedy: Functions and Naked Returns In Go](#)
- [Axel Wagner: The trouble with optional interfaces](#)
- [Banzai Cloud: Márk Sági-Kazár: Error handling practices in Go](#)
- [Build web application with Golang: 11.1 Error handling](#)
- [Carl M. Johnson: Creating Domain Specific Error Helpers in Go With errors.As](#)
- [Cloud Immunity: Kyle Quest: 50 Shades of Go: Traps, Gotchas, and Common Mistakes for New Golang Devs](#)
- [DEV Community: Lukas Lukac: How to handle errors in Go? \[5 rules\]](#)
- [Evil Martians: Sergey Alexandrovich: Errors in Go: From denial to acceptance](#)
- [freeCodeCamp: Bill Sourour: Code dependencies are the devil.](#)
- [Go Review: io: make clear that EOF should not be wrapped](#)
- [golang/go: Proposals: annotate which error values should never be wrapped](#)
- [hacker-laws: Hyrum's Law \(The Law of Implicit Interfaces\)](#)
- [Joe Shaw: Don't defer Close\(\) on writable files](#)

- [Joe Shaw: Error handling in Go HTTP applications](#)
- [Justinas Stankevičius: Best Practices for Errors in Go](#)
- [Linux Programmer's Manual: CLOSE\(2\)](#)
- [Maxime Vidori: GOLANG ERR SHADOWING](#)
- [Medium: Higher-Order Functions: Saurabh Nayar: Golang: Six Error Handling techniques to help you write elegant code](#)
- [Medium: NOP::Nuances of Programming: Olga Sayfudinova: Golang – изящная обработка ошибок](#)
- [OpenCredo: Andrew Morgan: What I Don't Like About Error Handling in Go, and How to Work Around It](#)
- [Orso Labs: Marco Molteni: Handling errors only once and logging them with full context in Golang](#)
- [Qvault.io: Lane Wagner: Top 6 Golang Logging Best Practices](#)
- [Oxeye.io: Gal Goldstein, Daniel Abeles: "ParseThru" – Exploiting HTTP Parameter Smuggling in Golang](#)
- [Sourcegraph: Marvin Guerra: GopherCon 2019 - Handling Go errors](#)
- [Reddit: r/golang: u/nesigma: Do we have to check for errors when we call Close\(\)?](#)
- [Reddit: r/golang: u/nesigma: Don't defer Close\(\) on writable files](#)
- [Stack Overflow: Dumitru: Check errors when calling http.ResponseWriter.Write\(\)](#)
- [Stack Overflow: johannes: Go method chaining and error handling](#)
- [Stack Overflow: Nathan H: Custom errors in golang and pointer receivers](#)
- [Stack Overflow: rcorre: Should I error-check Close\(\) on a response body?](#)
- [The Go Blog: Damien Neil and Jonathan Amsterdam: Working with Errors in Go 1.13: Whether to Wrap](#)
- [WEB3 COACH: Lukas Lukac: How to handle errors in Go? \[5 rules\]](#)
- [Wiki: Dependency hell](#)
- [yourbasic.org/golang: Shadowed variables](#)

