

# Пакет context в Go

Романов Никита Вячеславович



<https://github.com/capsulated/go-context-example>

# Что такое контекст

context.Context

Если мы взглянем на документацию к пакету context, то первый абзац будет таким:

“

Package context defines the Context type, which carries deadlines, cancellation signals, and other request-scoped values across API boundaries and between processes.

”

“

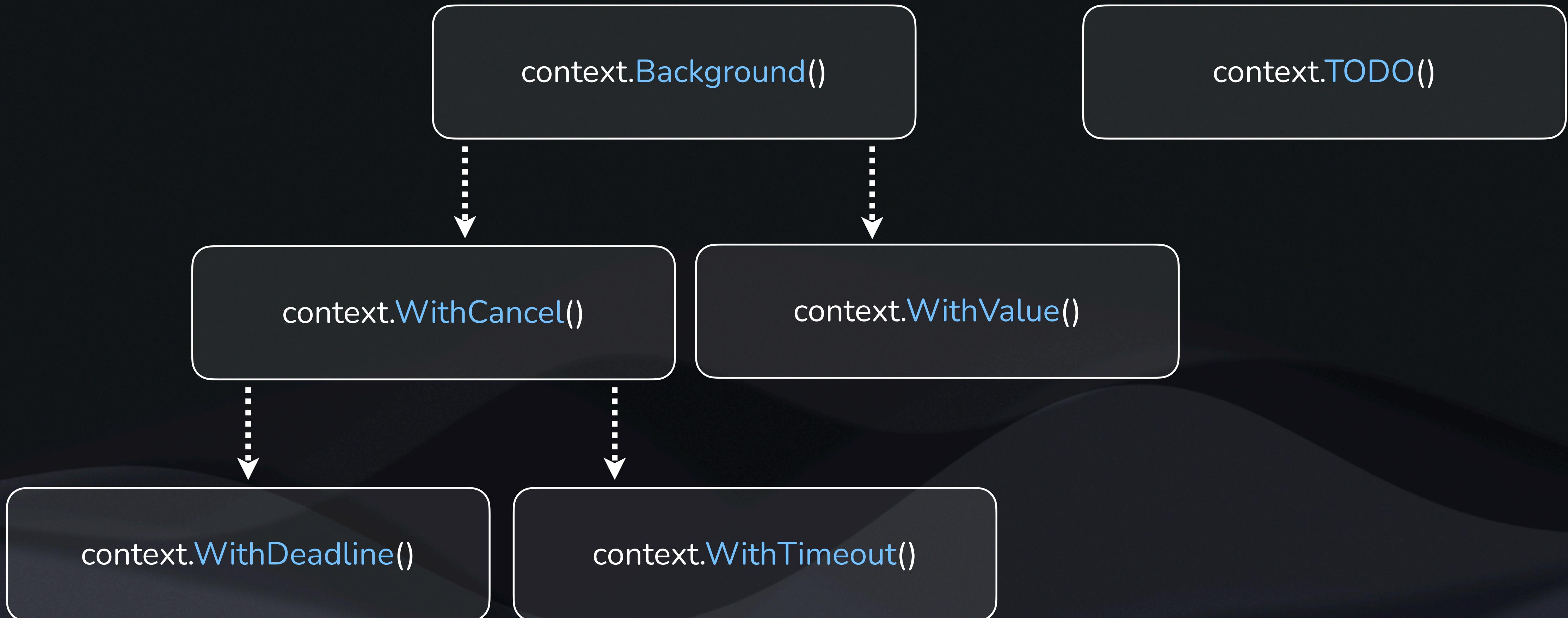
Пакет context определяет тип Context, который позволяет управлять дедлайнами, сигналами отмены и другими значениями области действия запросов между границами API и процессами.

”

Контекст — это объект, который предназначен в первую очередь для того, чтобы иметь возможность отменить извне выполнение потенциально долгой операции. Кроме того, с помощью контекста можно хранить и передавать информацию между функциями и методами внутри вашей программы.

# Виды контекстов

## Иерархия контекстов



# context.WithTimeout()

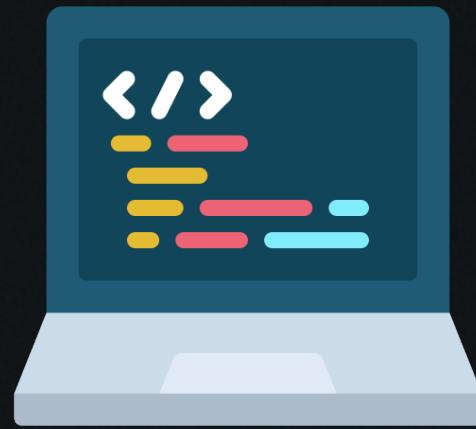
С примером приоритета таймаута



[https://github.com/capsulated/go-context-example/blob/main/cmd/01\\_priority/main.go](https://github.com/capsulated/go-context-example/blob/main/cmd/01_priority/main.go)

# context.WithCancel()

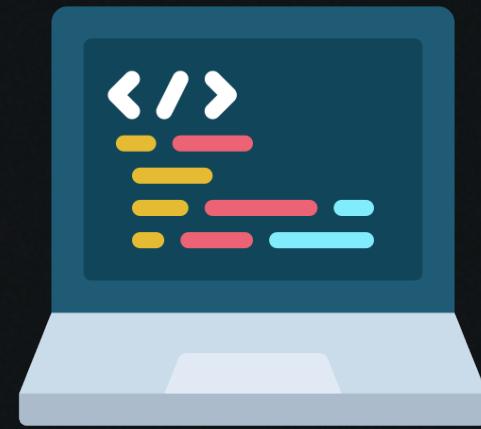
Контекст с отменой



[https://github.com/capsulated/go-context-example/blob/main/cmd/02\\_cancel/main.go](https://github.com/capsulated/go-context-example/blob/main/cmd/02_cancel/main.go)

# context.WithCancel()

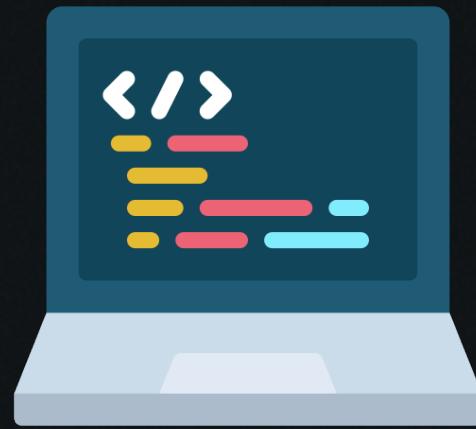
Контекст с отменой



[https://github.com/capsulated/go-context-example/blob/main/cmd/03\\_value/main.go](https://github.com/capsulated/go-context-example/blob/main/cmd/03_value/main.go)

# context.WithDeadline()

“Синтаксический сахар” на основе context.WithTimeout()

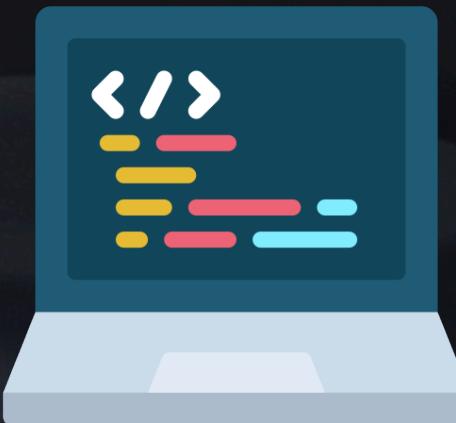


[https://github.com/capsulated/go-context-example/blob/main/cmd/04\\_deadline/main.go](https://github.com/capsulated/go-context-example/blob/main/cmd/04_deadline/main.go)

# context.withValue()



Передача данных через контекст является антипаттерном, поскольку это порождает неявный контракт между компонентами вашего приложения, к тому же ещё и ненадёжный.



[https://github.com/capsulated/go-context-example/blob/main/cmd/03\\_value/main.go](https://github.com/capsulated/go-context-example/blob/main/cmd/03_value/main.go)

# Когда использовать context

Когда необходимо использовать контекст

Сетевые запросы (API, веб-сервисы и т.п.)

Потенциально длительные горутины

# Best practice

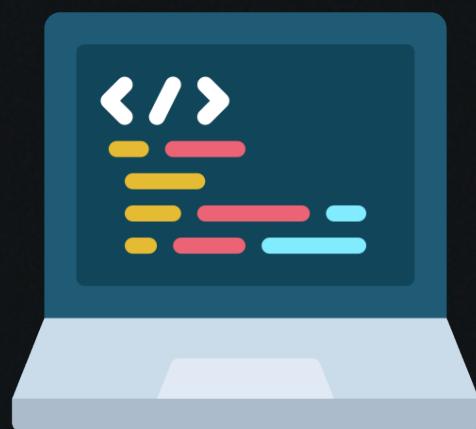
## Советы по использованию контекста

- Контекст всегда передаётся первым аргументом (общепринятое соглашение)
- Контекст передаётся только в функции и методы, и не хранится в состоянии объекта (к примеру, поле типа или глобальная переменная)
- context.WithValue используется только для передачи особых данных, типа id трассировки, дополнительных вспомогательных данных
- context.Background должен использоваться только как родительский
- Используйте context.TODO, если не ясно, какой использовать
- Не забывайте вызывать функцию отмены контекста
- Не передавайте функцию отмены
- Контроль за завершением контекста должен быть на уровне создания

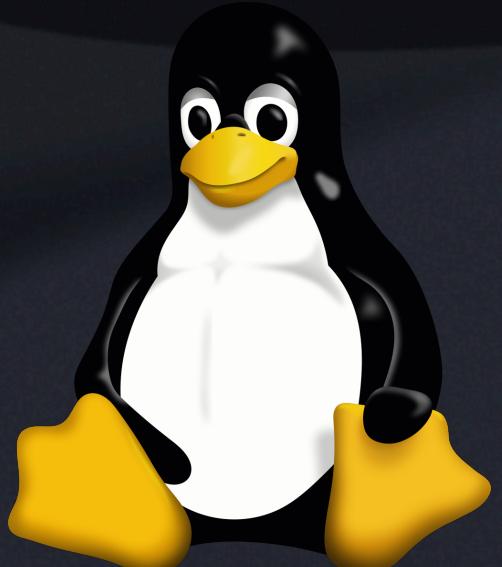
# Немного о Gracefully Shutdown

И как контекст используется в реализации этого паттерна

Gracefully Shutdown (изящное завершение программы) - функционал завершения выполнения процесса выполнения на основе обработки сигналов операционной системы.



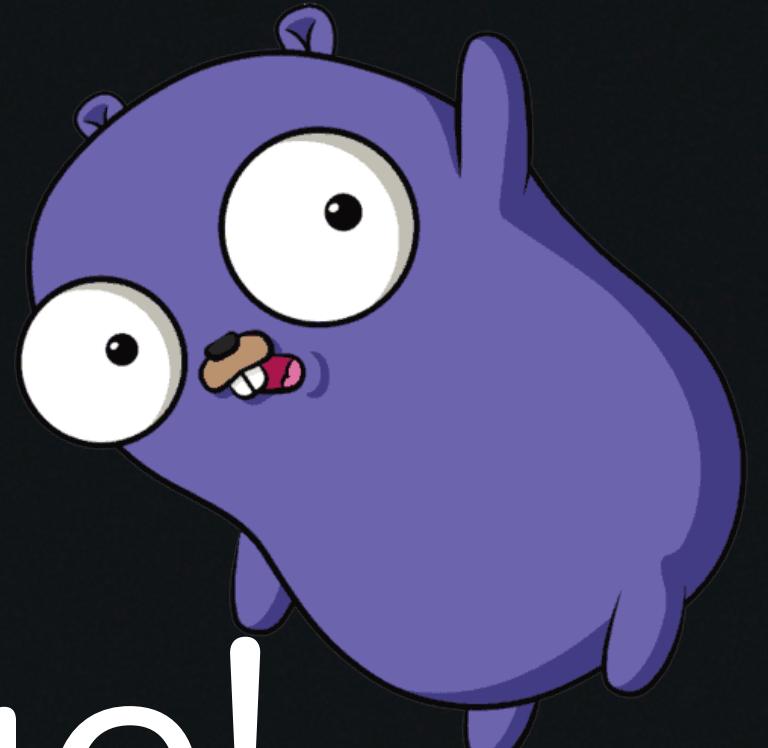
[https://github.com/capsulated/go-context-example/blob/main/cmd/06\\_shutdown/main.go](https://github.com/capsulated/go-context-example/blob/main/cmd/06_shutdown/main.go)



# Полезные ссылки

- [pkg.go.dev](#) / Документация к пакету context
- [The Go Blog](#) / Больше примеров от авторов языка
- [Go by Example](#) / Раздел о контекстах
- [Implementing Graceful Shutdown in Go](#) Leonidas Vrachnis / Статья
- [Golang для профи](#) Михалис Цукалос / Книга

# Спасибо за внимание!



Романов Никита Вячеславович