

Карта курса

Продвинутый

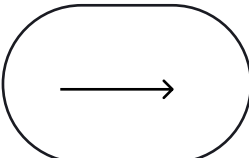
Go-разработчик

Купить курс

2 часа	20 часов	20 часов	20 часов	20 часов	40 часов
Введение и входной тест	Спринт 1 HTTP-приложения, время, контекст	Спринт 2 Обмен данными, хранение данных на диске	Спринт 3 Сжатие и шифрование данных, подключение базы данных, логирование ошибок.	Спринт 4 Обработка входящих и исходящих запросов в асинхронном режиме.	Спринты 5-6 Первый выпускной проект
20 часов	20 часов	20 часов	20 часов	20 часов	20 часов
Спринт 7 Паттерны проектирования, профилирование, стилизация, статический анализ кода	Спринт 8 Документация, кодогенерация, Go comment flags	Спринт 9 Расширенная стандартная библиотека: net, sync/x/sync, bytes, bufio	Спринт 10 Расширенная стандартная библиотека: math/rand, crypto/rand, вызов внешних приложений. Обзор остальной части стандартной библиотеки. gRPC	Спринт 11-12 Второй выпускной проект	

Чему вы научитесь на курсе:

- писать тесты и проверять функциональность и корректность кода;
- находить и исправлять синтаксические и стилистические ошибки кода;
- проектировать REST API;
- читать код на Go и понимать решаемую им задачу;
- проводить код-ревью приложений на Go;
- реализовывать архитектурные решения и паттерны проектирования на Go;
- портировать часть продакшен-кода с известного вам языка на Go под руководством более опытного разработчика;
- участвовать в проектировании архитектурных решений для новых сервисов на Go;
- внедрять в сервис на Go информативное и высокопроизводительное логирование.
- улучшать быстродействие уже написанного кода на Go;
- проектировать и писать микросервис самостоятельно;
- расширять функциональность существующего сложного сервиса;
- перекладывать продуктовые задачи в код на Go;
- писать продвинутые тесты.



Пакеты стандартной библиотеки

В этом модуле расскажем о популярных пакетах Go. Вы научитесь писать и тестировать HTTP-приложения с `net/http`, управлять передачей данных и временем жизни задач с `context`, делать запросы в базу данных, познакомитесь с особенностями сериализации и обработки ошибок в Go.

Проекты

Вам предстоит выбрать один из практических треков: «Go в веб-разработке» или «Go в DevOps». Практический трек связывает все задания курса в единый проект, который вы разрабатываете инкрементально до получения финального результата. Инкремент — это часть кода, которая добавляет новые свойства или функции вашему сервису. На треке «Go в веб-разработке» вы будете писать сервис для сокращения URL. А на треке «Go в DevOps» — сервис для сбора метрик и алертинга.

Спринт 1

2 недели
20 часов

Содержание уроков главы:

`time, context`

- время: дата, интервалы, таймеры
- контекст: отмена операций и передача данных

`net/http`

- структура проекта
- HTTP-клиент
- HTTP-сервер
- тестирование HTTP-приложений
- HTTP-библиотеки

Практика:

Практический инкремент* 1:

Ваш код на данном этапе практического трека позволяет раздавать данные клиентам по протоколу HTTP.

Практический инкремент 2:

Код не изменяет функциональности, добавляются тесты для проверки корректности работы системы.

Практический инкремент 3:

Код позволяет более гибко отправлять и принимать данные.

Спринт 2

2 недели
20 часов

Содержание уроков главы:

`encoding`

- структурные теги
- JSON
- XML
- Gob
- стандартные сериализаторы
- сторонние сериализаторы

`os`

- переменные окружения
- работа с файлами

`flag`

- аргументы командной строки

Практика:

Практический инкремент 4:

Код использует JSON как универсальный формат обмена данными.

Практический инкремент 5:

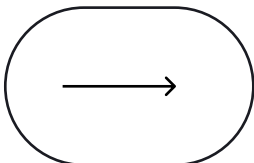
Код использует переменные окружения ОС для установки состояния сервиса на момент запуска.

Практический инкремент 6:

Код использует файлы для персистентного хранения данных на диске.

Практический инкремент 7:

Код использует аргументы командной строки вместо переменных окружения ОС.



Спринт 3

2 недели
20 часов

Содержание уроков главы:

compress

- стандартные методы сжатия
- сторонние библиотеки и форматы

hash и crypto

- хеширование и шифрование

database/sql

- интерфейсный подход и драйверы
- запросы к БД
- запись в БД, состояния и транзакции

errors, log

- интроспекция и логирование ошибок

Практика:

Практический инкремент 8:

Код умеет сжимать данные ответа с помощью gzip.

Практический инкремент 9:

Код умеет использовать пакет crypto для шифрования данных.

Практический инкремент 10:

Код не изменяется. Студент выбирает знакомую или удобную ему БД, драйвер к ней и проектирует схему БД.

Практический инкремент 11:

Код умеет делать запросы к БД и использует её в качестве персистентного хранилища.

Практический инкремент 12:

Код умеет эффективно совершать сложные запросы к БД.

Практический инкремент 13:

Код умеет определять типы ошибок и логировать события в системе.

Модуль 2

Конкурентность

Расскажем, что такое многопоточность на уровне операционной системы. Вы узнаете, в чём разница между потоком и процессом, какие существуют виды многопоточности, что происходит при переключении потоков, как бороться с deadlock и data race. Отвечая на эти и другие вопросы, плавно перейдём к устройству планировщика Go. Потом познакомим с легковесными потоками в языке (goroutines) и другими средствами для написания многопоточных приложений.

Спринт 4

2 недели
20 часов

Содержание уроков главы:

Многопоточность в Go

- основы многопоточности
- горутины
- каналы
- получение данных из каналов
- синхронизация состояния
- паттерны многопоточности

Практика:

Практический инкремент 14:

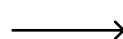
Код умеет обрабатывать входящие и исходящие данные в асинхронном режиме.

Спринт 5-6

4 недели
40 часов

Первый выпускной проект

Вы сможете выполнить проект по предложенному нами ТЗ индивидуально или в команде с другими студентами. Также вы сможете сделать свой проект, предварительно согласовав его с ментором. Проверять выпускные проекты будут менторы. На выполнение проекта у вас будет 4 недели.



Модуль 3

Спринт 7

1 неделя
6 часов

Модуль 4

Спринт 7

1 неделя
14 часов

Спринт 8

2 недель
20 часов

Паттерны проектирования

В этом модуле рассмотрим паттерны проектирования и антипаттерны программирования на Go. Вы сможете выбирать наиболее подходящую архитектуру для вашего сервиса.

Содержание уроков главы:

Паттерны проектирования на Go

- порождающие паттерны
- структурные паттерны
- поведенческие паттерны

Антипаттерны программирования на Go

- антипаттерны программирования на Go

Тулинг

Сильная сторона Go — встроенные инструменты разработчика. В этом модуле вы научитесь использовать тулинг для стилизации и статического анализа кода, создания документации, кодогенерации и профилирования.

Содержание уроков главы:

Профилирование

- pprof
- сторонний тулинг

Стилизация кода

- gofmt и goimports

Статический анализ кода

- go vet
- go/ast
- x/analysis
- staticcheck

Содержание уроков главы:

Документация

- godoc и swagger
- example_test.go

Кодогенерация

- кодогенерация проектов
- кодогенерация как замена дженерикам
- генерация с помощью шаблонизации

Go comment flags

- go-flags, go:flags, build flags
- ldflags
- флаги и компиляции
- импорты



Модуль 5

Расширенная стандартная библиотека

В Go многие вещи доступны «из коробки», и приветствуется написание собственных решений на основе стандартной библиотеки. В этом модуле вы завершите знакомство со встроенными пакетами Go, научитесь писать более эффективный и лаконичный код.

Спринт 9

2 недели
20 часов

Содержание уроков главы:

net

- работа с TCP/UDP
- IP адреса

sync и x/sync

- конкурентный map с sync.Map
- ожидание выполнения асинхронных задач
- одноразовое выполнение кода

bytes

- буфер байт
- блочное чтение байт

bufio

- bufio.Scanner
- эффективная работа с текстом большого размера

Спринт 10

2 недели
20 часов

Содержание уроков главы:

math/rand и crypto/rand

- генератор случайных чисел
- криптографический генератор случайных чисел

OS

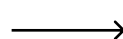
- работа с директориями (траверсинг, walk-функции)
- вызов внешних приложений
- сигналы

Обзор остальной части стандартной библиотеки

- stdlib и golang.org/x
- sort
- websocket
- trace
- heap
- list

gRPC

- gRPC и тесты



Второй выпускной проект

Вы сможете выполнить проект по предложенному нами ТЗ индивидуально или в команде с другими студентами. Также вы сможете сделать свой проект, предварительно согласовав его с ментором. Проверять выпускные проекты будут менторы. На выполнение проекта у вас будет 4 недели.

Вебинары

В дополнение к теории и практике менторы будут проводить для вас вебинары один раз в спринт, в конце первой недели. Вебинары нужны для того, чтобы подробнее остановиться на сложных темах, разобрать самые частые ошибки и ответить на вопросы по курсу, Go или программированию в целом.

Сессии 1:1

У каждого студента будет возможность один раз в спринт созвониться со своим ментором один на один. В отведенные полчаса вы можете не только задать вопросы по проекту, но и обсудить карьерные ожидания, поделиться опытом разработки или затронуть любую другую интересующую вас тему.