Продвинутый Go-разработчик

6 месяцев

продолжительность курса

Купить курс

Нажмите, чтобы перейти на страницу покупки

2 ЧАСА

00

Введение
и входной тест

3 НЕДЕЛИ | 34 ЧАСА

01

Структура проекта, работа с HTTP, аргументами командной строки, переменными окружения

2 НЕДЕЛИ | 33 ЧАСА

02

Логирование, обмен данными, сжатие данных, хранение данных на диске

2 НЕДЕЛИ | 30 ЧАСОВ

03

Отмена операций, управление временем выполнения, подключение базы данных, интроспекция ошибок

2 НЕДЕЛИ | 23 ЧАСА

04

Шифрование данных, обработка входящих и исходящих запросов в асинхронном режиме

5 НЕДЕЛЬ | 70 ЧАСОВ

05

Первый итоговый проект курса «Продвинутый Go-разработчик»

2 НЕДЕЛИ | 30 ЧАСОВ

06

Паттерны проектирования, антипаттерны программирования, профилирование, стилизация, документация

2 НЕДЕЛИ | 30 ЧАСОВ

07

Статический анализ кода, дженерики и кодогенерация, флаги сборки и компиляции

2 НЕДЕЛИ | 30 ЧАСОВ

08

Экспресс-обзор стандартной библиотеки, генерация случайных чисел, чтение данных и буфер, работа с операционной системой

2 НЕДЕЛИ | 32 ЧАСА

09

Примитивы синхронизации, работа с сетью, protocol buffers и gRPC

5 НЕДЕЛЬ | 70 ЧАСОВ

10

Второй итоговый проект курса «Продвинутый Go-разработчик»

2 НЕДЕЛИ | 15 ЧАСОВ | ТАРИФ

11

Брокеры сообщений и асинхронные паттерны

3 НЕДЕЛИ | 20 ЧАСОВ | ТАРИФ

12

Observability — логирование, метрики, трассировка

3 НЕДЕЛИ | 23 ЧАСА | ТАРИФ

13

Kubernetes и деплой Go-сервисов

Что вас ждёт на курсе

Вебинары

Один раз в спринт менторы будут проводить вебинары. Они будут дополнять теоретический материал и помогут разобраться с кейсами. Кроме того, это отличная возможность провести работу над ошибками и узнать ответы на волнующие вас вопросы по курсу, Go и программированию в целом.

Чему вы научитесь на курсе Продвинутый Go-разработчик

- Писать тесты и проверять функциональность и корректность кода
- Проектировать REST API
- Читать код на Go и понимать решаемую им задачу
- Проводить код-ревью приложений 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
- Проектировать и писать микросервис самостоятельно
- Перекладывать продуктовые задачи в код на Go
- Внедрять в сервис на Go информативное и высокопроизводительное логирование
- Улучшать быстродействие уже написанного кода на Go
- Реализовывать архитектурные решения и паттерны проектирования на Go
- Находить и исправлять синтаксические и стилистические ошибки кода
- Расширять функциональность существующего сложного сервиса
- Писать продвинутые тесты

Пакеты стандартной библиотеки 01

В этом модуле расскажем о популярных пакетах Go. Вы научитесь писать и тестировать HTTP-приложения, управлять временем выполнения операций, делать запросы к базе данных и обрабатывать ошибки. Также вы познакомитесь с особенностями логирования, чтения и записи в файл, сериализации и десериализации данных.

Проекты

Вам предстоит выбрать один из практических треков: «Сервис сокращения URL» или «Сервис сбора метрик и алертинга». 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.

Содержание модуля

№	Продолжительность	Теория	Практика
Спринт 1	3 недели | 34 часа	**Пакет net/http. Работа с HTTP** • Структура проекта • Создание HTTP-сервера • Тестирование HTTP-приложения • Использование HTTP-клиента • Выбор HTTP-библиотеки **Пакет flag. Чтение аргументов командной строки** • Аргументы командной строки **Пакет os. Получение переменных окружения** • Переменные окружения	**Инкремент 1** Ваш код на данном этапе практического трека позволяет раздавать данные клиентам по протоколу HTTP. **Инкремент 2** Код не изменяет функциональности, добавляются тесты для проверки корректности работы системы. **Инкремент 3** Код позволяет более гибко отправлять и принимать данные. **Инкремент 4** Код использует аргументы командной строки. **Инкремент 5** Код использует переменные окружения ОС для установки состояния сервиса на момент запуска.

Содержание модуля

№	Продолжительность	Теория	Практика
Спринт 2	2 недели | 33 часа	**Пакет log. Логирование в приложении** • Стандартные и сторонние пакеты для логирования • Логирование через middleware **Пакет encoding. Сериализация и десериализация данных** • Основы REST API • Структурные теги • Стандартные сериализаторы • Сторонние сериализаторы **Пакет compress. Сжатие данных** Оптимизация передачи данных **Пакет os. Операции с файлами** • Чтение и запись в файл	**Инкремент 6** Код умеет логировать события в системе. **Инкремент 7** Код использует JSON как универсальный формат обмена данными. **Инкремент 8** Код умеет сжимать данные ответа с помощью gzip. **Инкремент 9** Код использует файлы для персистентного хранения данных на диске.
Спринт 3	2 недели | 30 часов	**Пакеты time, context. Отмена операций и управление временем выполнения** • Время: даты, интервалы, таймеры • Контекст: отмена операций **Пакет database/sql. Взаимодействие с базами данных SQL** • Пакет gomock: имитация баз данных • Абстрактный интерфейс и SQL-драйверы • Запросы к базе данных • Запись в базу данных **Пакет errors. Обработка ошибок** • Интроспекция ошибок	**Инкремент 10** Участник курса подключает сервис к СУБД PostgreSQL и проектирует схему базы данных. **Инкремент 11** Код умеет делать запросы к БД и использует её в качестве персистентного хранилища. **Инкремент 12** Код умеет эффективно совершать сложные запросы к БД. **Инкремент 13** Код умеет определять типы ошибок.
Спринт 4	2 недели | 23 часа	**Пакеты hash, crypto. Безопасность информации** • Хеширование и шифрование • Авторизация: JSON Web Token	**Инкремент 14** Код умеет использовать пакет crypto для шифрования данных.

Конкурентность 02

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 datarace. Отвечая на эти и другие вопросы, плавно перейдём к устройству планировщика Go. Потом познакомим вас с легковесными потоками в языке (goroutines) и другими средствами для написания многопоточных приложений.

Содержание модуля

№	Продолжительность	Теория	Практика
Спринт 4	2 недели | 23 часа	**Многопоточность** • Основы многопоточности • Многопоточность в Go • Каналы • Паттерны многопоточности	**Инкремент 15** Код умеет обрабатывать входящие и исходящие данные в асинхронном режиме.
Спринт 5	5 недель | 70 часов	**Первый итоговый проект** Вы напишете итоговый проект по предложенному нами ТЗ или реализуете собственную идею, предварительно согласовав её с ментором. Работать можно индивидуально или в команде с другими участниками курса. Проверять итоговые проекты будут менторы. На выполнение проекта отводится 4 недели, и ещё 1 неделя на проверку.	

Паттерны проектирования 03

В этом модуле рассмотрим паттерны проектирования и антипаттерны программирования на Go. Вы узнаете, как реализовать наиболее распространённые паттерны проектирования приложений и как избежать неявных ошибок при программировании на Go.

Содержание модуля

Спринт 6	1 неделя | 15 часов	**Паттерны проектирования на Go** • Порождающие паттерны • Структурные паттерны • Поведенческие паттерны **Антипаттерны программирования на Go** • Постулаты Go • Лучшие практики, антипаттерны

Тулинг

04

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 модуля

№	Продолжительность	Теория	Практика
Спринт 6	1 неделя | 15 часов	**Профилирование** • Бенчмарки • Инструмент pprof **Стилизация** • Форматирование кода: gofmt и goimports **Документация** • Генерирование документации командой godoc. Спецификация Swagger • Шаблон example_test.go	**Инкремент 16** Код исправлен согласно статистике, собранной с помощью pprof **Инкремент 17** Код отформатирован с помощью gofmt или goimports. **Инкремент 18** Экспортированные функции кода описаны комментариями в формате godoc.
Спринт 7	2 недели | 30 часов	**Статический анализ кода** • Команда go vet • Пакет go/ast • Пакет x/analysis • Пакет staticcheck **Дженерики и кодогенерация** • Кодогенерация • Дженерики **Флаги сборки и компиляции** • Флаги сборки и компиляции: build constraints	**Инкремент 19** Код проверен статическим мультираннером и не содержит ошибок. **Инкремент 20** Код добавляет версию и другие метаданные при компиляции.

Расширенная стандартная библиотека

05

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 и познакомитесь с «расширенной» стандартной библиотекой языка, в которой содержится большое количество полезных пакетов.

Содержание модуля

№	Продолжительность	Теория	Практика
Спринт 8	2 недели | 30 часов	**Экспресс-обзор стандартной библиотеки** • Пакеты стандартной библиотеки (stdlib) • Расширенная стандартная библиотека (golang.org/x) **Генерация случайных чисел** • Пакеты math/rand и crypto/rand **Чтение данных и буфер** • Пакет bytes • Пакет bufio **Работа с операционной системой** • Пакет os. Работа с директориями и процессами • Вызов внешних приложений, сигналы	**Инкремент 21** Код умеет использовать асимметричное шифрование. **Инкремент 22** Код умеет работать с файлами конфигов. **Инкремент 23** Код умеет реагировать на сигналы ОС.
Спринт 9	2 недели | 30 часов	**Примитивы синхронизации** • Пакеты sync и x/sync **Работа с сетью** • Пакет net. Работа с TCP и UDP • IP-адреса **Protocol buffers и gRPC** • Protocol buffers и gRPC • Разработка gRPC-сервера и клиента	**Инкремент 24** Код умеет обрабатывать запросы с учётом IP-адреса клиента. **Для сервиса сокращения URL:** ограничивает доступ к некоторым хендлерам только для запросов из внутренней сети. **Для сервиса сбора метрик:** сервер принимает метрики только от доверенных IP-адресов, если выставлен специальный флаг. **Инкремент 25** Код умеет обрабатывать запросы по протоколу gRPC.

Второй итоговый проект

№	Продолжительность	Содержание
Спринт 10	5 недель | 70 часов	**Второй итоговый проект** Вы напишете итоговый проект по предложенному нами ТЗ или реализуете собственную идею, предварительно согласовав её с ментором. Работать можно индивидуально или в команде с другими участниками курса. Проверять выполненные участниками курса проекты будут менторы. На выполнение проекта отводится 3 недели, и ещё 1 неделя на проверку.

Модуль в тарифе курса «Продвинутый Go-разработчик + инфраструктура и продакшн»

В этом модуле вы отработаете ключевые продакшн-навыки: развёртывание, масштабирование и наблюдаемость Go-сервисов.

Содержание модуля

№	Продолжительность	Содержание
Спринт 11	2 недели | 15 часов	**Брокеры сообщений и асинхронные паттерны** • Что такое брокеры сообщений и зачем они нужны • Основы работы с RabbitMQ: продюсеры и консюмеры • Очереди, exchange, routing keys, ack/retry механизмы • Основы Kafka: топики, партиции, offset, consumer groups • Использование segmentio/kafka-go или confluent-kafka-go • Идиомы event-driven архитектуры: pub/sub, fan-out, DLQ • Интеграция брокеров в микросервисы • Отладка, мониторинг и тестирование брокеров
Спринт 12	3 недели | 20 часов	**Observability — логирование, метрики, трассировка** • Введение в Observability: Logs, Metrics, Traces • Структурированное логирование в Go • Интеграция с OpenTelemetry • Сбор метрик с помощью prometheus/client_golang • Настройка Prometheus и Grafana для визуализации • Трейсинг запросов • Интеграция observability-инструментов в CI/CD

Содержание модуля

№	Продолжительность	Содержание
Спринт 13	3 недели | 23 часа	**Kubernetes и деплой Go-сервисов** • Введение в контейнеризацию: Docker и основы DevOps-цикла • Dockerfile для Go: best practices, multi-stage сборка • Работа с Docker Compose: локальная разработка с зависимостями (PostgreSQL, Redis) • Основы Kubernetes • Helm-чарты: шаблонизация и переиспользуемость манифестов • Настройка деплоя Go-приложения в кластер Kubernetes • CI/CD для Go-сервисов

Дополнительный модуль: Алгоритмы и структуры данных 06

В этом модуле вы узнаете, как работать с самыми популярными структурами данных с помощью языка Go: хеш-таблицами, связными списками и графами.

Содержание модуля

№	Продолжительность	Теория
Спринт 13	Введение в алгоритмы 27 часов	• Понятие алгоритма • Понятие сложности • Временная и пространственная сложность • Алгоритмические собеседования
Спринт 14	Основные структуры данных 20 часов	• Массив, связный список, стек, очередь • Сложность операций вставки, поиска и удаления • Представление данных в памяти • Пространственная сложность алгоритма
Спринт 15	Рекурсия и сортировки 20 часов	**Рекурсия** • Понятие рекурсии • Принцип «разделяй и властвуй» • Бинарный поиск **Сортировки** • Квадратичные сортировки • Сортировка слиянием • Быстрая сортировка • Линейная сортировка подсчётом
Спринт 16	Хеш-функции и хеш-таблицы 20 часов	• Абстракция отображения • Понятие и свойства хеш-функции, примеры • Структура данных хеш-таблица • Коллизии и способы их разрешения

Медведев Кирилл
Директор АНО ДПО «Образовательные технологии Яндекса»