Практикум

Продолжительность курса – 2-3 месяца

Карта курса Go-разработчик в своём темпе. Блок 1

Купить курс

2 часа	30 часов	30 часов	30 часов	30 часов
Входной тест и введение	**Спринт 1** HTTP-приложения, время, контекст	**Спринт 2** Обмен данными, хранение данных на диске, сжатие данных	**Спринт 3** Шифрование данных, подключение базы данных, логирование ошибок	**Спринт 4** Обработка входящих и исходящих запросов в асинхронном режиме

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.

Модуль 1

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: «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.

Спринт 1

2 недели
30 часов

Теория

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

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.

Спринт 2

2 недели
30 часов

Теория

encoding

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

os

- переменные окружения
- работа с файлами

flag

- аргументы командной строки

compress

- сжатие данных

Практика

Инкремент 4

Код использует JSON как универсальный формат обмена данными.

Инкремент 5

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

Инкремент 6

Код использует файлы для персистентного хранения данных на диске.

Инкремент 7

Код использует аргументы командной строки вместо переменных окружения ОС.

Инкремент 8

Код умеет сжимать данные ответа с помощью gzip.

Спринт 3

2 недели
30 часов

Теория

hash и crypto

- хеширование и шифрование

database/sql

- пакет gomock, имитация данных для тестирования
- обобщённый подход и драйверы
- запросы к базе данных
- запись в базу данных, SQL-инструкции и транзакции

errors, log

- интроспекция и логирование ошибок

Практика

Инкремент 9

Код умеет использовать пакет crypto для шифрования данных.

Инкремент 10

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

Инкремент 11

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

Инкремент 12

Код умеет эффективно совершать сложные запросы к БД.

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 datarace. 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
30 часов

Теория

Многопоточность

- основы многопоточности
- горутины
- каналы
- синхронизация состояния
- паттерны многопоточности

Практика

Инкремент 14

Код умеет обрабатывать входящие и исходящие данные в асинхронном режиме.