

Карта курса

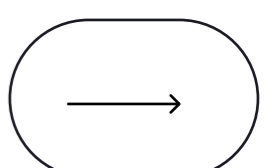
Go-разработчик

[Купить курс](#)

2 часа	20 часов	20 часов	20 часов	20 часов	40 часов
Входной тест и введение	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 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.



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.

Задания будут уложены в практические треки. Практический трек связывает все задания в единый проект, который вы разрабатываете инкрементально до получения финального результата. Инкремент — это часть кода, которая добавляет новые свойства или функции вашему сервису. На курсе есть два практических трека: «Go в веб-разработке» и «Go в DevOps». Вам предстоит выбрать один из них.

Спринт 1

2 недели
20 часов

Теория

`time, context`

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

`net/http`

- HTTP-клиент
- HTTP-сервер
- тестирование HTTP-приложений
- HTTP-библиотеки

Практика

Практический инкремент 1

Код студента 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

Теория

`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

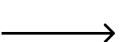
Код использует переменные окружения ОС для установки состояния сервиса на момент запуска.

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

Теория

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 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
20 часов

Теория

Многопоточность в Go

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

Практика

Практический инкремент 13

Код умеет обрабатывать входящие и исходящие данные в асинхронном режиме.

Спринты 5, 6

4 недели
40 часов

Выпускной проект

Вы выберете выпускной проект из предложенных вариантов или придумаете собственный, в этом случае его одобрит ваш ментор. Вы сможете объединиться в группы с другими студентами или работать индивидуально. Выпускной проект проверит ментор. На выполнение выпускного проекта у вас будет 4 недели.