SQLabs
המכללה הגבוהה להייטק

Advanced Go (16 hours)

Overview

Google’s Go Language is one of the fastest grown programming languages in the world and with good reason. It is a modern language designed from the ground up by some of the most famous names in Computer Science. Go leverages the capabilities of modern multi core architectures and computer language innovations to be o extremely fast, efficient and maintainable.

This course, designed for developers who already have a basic grasp of Google's GO language takes you beyond the Go Playground, into some more advanced yet varied practical topics. It includes lectures and is practically oriented with a great deal of hands on practice and exercises.

Intended Audience

Developers with experience with the GO language who want deeper knowledge of the internals of the Go language and advanced features to enable more effective Go development.

Prerequisites

Practical experience with programming in the GO programming language

Course Contents

Effective Go:

- Formatting, comments and naming conventions
- Switch, fallthrough and no ops
- Redeclaration and Reassignment
- Range with Unicode
- Defer best practices
- New vs Make vs Constructors
- Arrays by pointer vs slice
- Implementation of append
- Global init functions
- Web serving int counter
- Channels blank identifier importing
- Using the blank identifier in importing
- Embedding: Interfaces, Structs vs Inheritance

Go By Example:

- Variadic Functions
- Error handling best practices
- Channels: Directions, Timeouts,
- Synchronizing: WaitGroup, Atomic, Mutex, Channels
- Custom Sorting
- Panic ,Recover, Rethrow Idiom
- Writing Collection functions
- Marshalling/Unmarshalling JSON
- C/go Interop
- Go Reflection

Web services in Go:

- Contexts and Cancellation
- HTTP Clients
- HTTP Servers
- Serving REST with Gorilla

Go and Software Development

- Object Oriented Principles and Go:
- The internals of Go Interfaces
- Type conversion
- Benefits of DI
- DI in Go
- Classic OO Design Patterns in GO?
- Go Idioms/ Patterns (Reader , Writer)
- Idiomatic Go and Antipatterns
- Unit and Service Tests
- Custom/Handwritten mocks
- Mocking frameworks gomock vs testify vs mockery
- Coverage, Go tool cover,
- Writing good tests / when coverage is not enough
- Code analyzers: lint, go report card