Advanced Golang

Duration : 4 Days

Prerequisites

Participants must be familiar with the following:

- Pointers
- Error Handling
- Functions
    - Variadic Functions
    - Anonymous Functions
    - Higher Order Functions
    - Closures
    - Deferred Functions
- Panic & Recovery
- Structs & Struct Composition
- Struct Methods

Outline

DAY-01

Topic #1: Quick Review

- Values, pointers
- Slices, arrays, maps

Topic #2: Memory In Go

- Stack & Heap
- What's a stack, what's a heap
- What goes on which
- Heap analysis using "-gcflags -m"

Topic #3: Interfaces

- What's an interface?
- Usage examples
- Interface Composition
- Empty Interfaces & Type Assertion

Topic #4: Modules & Packages

- Modules Overview
- Module Commands
- Packages Overview

- Private & Public Scope
- Package Initialization
- Nested Packages
- Using 3rd Party Modules

Topic #5: Generics

- What are Generics?
- Type Parameters
- Type Constraints
- Generic Functions
- Generic Data Structures

Topic #6: Cross Platform Support

- Compiling Cross Platform Binaries
- Build Flags

DAY-02 & 03

Topic #1: Concurrency Overview

- Goroutines
- Channels
- Buffered and Unbuffered channels
- Passing data in and out of channels
- Handling race conditions
- Using sync package
- Synchronizing goroutines using waitGroups
- Concurrent safe operations using sync/atomic package
- Parallelism in Go
- Cancel Propagation using Context

Topic #2: Advanced Concurrency

- Advanced Concurrency Patterns
- Runner
- Worker
- Pools
- Signals

DAY-03 & 04

Topic #1: HTTP Services

- HTTP package

- Building a web server
- Handling Requests and Responses
- Serving static files
- Serving JSON responses
- Creating & Using Middlewares
- Implementing Graceful Shutdown
- Exposing prometheus metrics

Topic #2: GRPC Services

- Protocol Buffers
- Communication Patterns in GRPC
- Request & Response
- Client Streaming
- Server Streaming
- Bidirectional Streaming
- Interceptors

DAY-04

Topic #1: Testing & Benchmarking

- Writing unit test cases
- Writing testable code
    - Structuring your code
- Benchmarking support in Go
    - Using benchstat

Topic #2: Performance Engineering

- API Load testing
- Diagnostics
    - Profiling
        - CPU
        - Heap memory
        - Mutex & Block
- Tracing
- Runtime statistics and events
- Garbage collection