**Advanced Go Progamming**

Topics

**Introduction**

* Review of Go Syntax
* Working with Built-in Types in Go
* Learning the Features of Advanced Go
* Working with the Advanced Go Build Tool
* Using Maps as Sets
* Working with Interface Keys and Slices
* Implementing Collection and Reduction
* Using the Pprof Tool
* Learning How to Reduce Garbage
* Working with Methods and Interfaces

**Synchronization**

* Atomicity
* Advanced Channels
* Cancellations
* Work stealing
* Error Propagation

**Files**

* JSON
* HTML
* XML

**Networking**

* Sockets
* TCP
* HTTP
* Data serialization

**Cryptography**

* Public / Private Key
* Encryption
* Hashing
* Digital Signatures

**Introspection**

* Review
* Settable

**Memory**

* Memory management
* Pointers
* Make
* New
* Parameters
* Returns

**Unsafe**

* Alignment
* Offset
* Unsafe pointers

**Tips and Tricks**

* Anonymous structures
* Performance profiling
* Mocking
* Method expressions

**Advanced**

* Modifying design patterns to work in Go – Go code smells.
* Go workspace and configuration idioms.
* Network and web programming with Go standard libraries.
* String, text, unicode processing and I/O with Go standard libraries.
* Compression and encryption with Go standard libraries.
* Database management with Go Standard libraries.
* Top external packages for testing (eg. testify, gingko)
* Top external packages for functionality (eg. go-set, go-ethereum, gorm)
* Top external packages for web programming (eg. gin, traefik, etv)
* External packages for microservices, web services and cloud computing.
* Synchronized rest template (equivalent)
* Inter service commn kafka/rabbitmq (asynchronous)
* Go Microservices (Gateway, registry) async call
* Docker and Kubernetes
* Mocking in Unit Tests
* Using Concurrency to Keep Data Consistent
* Using User-Defined Types
* Optimizing Code for Speed
* Using Idiomatic Go
* Working with Time, Errors, Common Interfaces, and Comments
* Working with and Managing Packages
* Creating a Scalable Work System
* Working with Databases in Go
* Serving a JSON API
* Testing and Building in Go

Use case from networking

Design Pattern specific to golang

Additional Topics

Golang basics