

## Go fundamentals

Develop backends and web services in Go

3jours / 21h

### Target audience

- Mastering Go fundamentals
- Implement a command line interface
- Produce logs
- Create HTTP services
- Access to databases
- Dockerize and deploy a Go Application

### Prerequisites

Backend or Full Stack developers.

### Course Outline

#### Module 1 : Introduction

- Presentation, organization of the course
- Go: Key-concepts, context, history
- Workstations conformity to prerequisites
  - Attendees workstation installation checks
  - Completing installations if needed
- Cloning git repository and preparing workspace

## **Module 2 : Fundamentals**

- Compilation, execution
- Packages, imports
- Types, variables, constants
- Functions
- Conditional constructs
- Iterations

Labs:

- Exercise on functions and conditional constructs
- Exercise on iterations

## **Module 3: Structures and pointers**

- Defer/panic/recover
- Structures
- Pointers

Labs:

- Mind agility game about pointers and structures

## **Module 4 : Fundamental aggregating types**

- Arrays
- Slices
- Maps
- Iterations over such types
- Closures

Labs:

- Arrays and slices manipulations
- Closures

## **Module 5 : Methods and interfaces**

- Methods and receivers
- Interfaces & Duck Typing
- The Empty interface
- Type assertions and type switches

Labs:

- Methods with struct receiver
- Type switches and recover

## **Module 6: Concurrent programming**

- Goroutines
- Channels
- Iterations on channels
- (non/) blocking I/O over channels
- Select : multiplexing

Labs:

- Goroutines
- Using channels for goroutines synchronization

## **Module 7 : Tooling**

- Go tooling (run, build, test, ...)
- The makefile
- Vendors
- Presentation of the training project

## **Module 8 : CLI : Command Line Interface**

- Running an application via the command line
- Native and libraries
- CLI Arguments parsing

Labs:

- Completing the CLI management of the training application

## **Module 9 : Logging**

- Logging in the application
- Native and libraries
- Configuring the logger
- Exploiting logs

Labs:

- Producing logs in the project application

## **Module 10 : Concurrency, multi-threading, unit tests**

- Go solution (routines, channel)
- Channels multiplexing (select)
- Tests

Labs:

- Implementing a request counter

### **Module 11 : Data access**

- Data access layers architecture
- Description of the data model
- Interface, DAO and Factory
- Testing and Mocking

Labs:

- Completing the model layer of the application
- Complete the DAO factory and the Mock implementation
- MongoDB ou PostgreSQL implementation

### **Module 12: Handling HTTP requests**

- Go web packages: native and libraries
- Layered architecture of the web
- Middlewares
- Routage and Endpoints
- Web tests

Labs:

- Completing the “web” package of the application

### **Module 13: Build and deployment**

- Dockerfile of a go executable
- Running the executable and the DB system (compose)
- Testing the service