

Updated 06/09/2024

Sign up

Formation Go

3 days (21 hours)

Presentation

Our [Go](#) training course will help you make large-scale programming easier and faster. Created by Google in 2009 and inspired by C and Pascal, Go is the preferred programming language for web applications and software.

In this course, we'll introduce you to Go and the basics of the language, then explain concurrency and unit testing so you can create and optimize your own package.

You'll also learn about variables, data types, functions, control structures and error handling. You'll discover GO's unique features, such as goroutines and channels, which enable concurrent programming and efficient communication between processes.

Like all our training courses, this one will introduce you to the latest stable version and its new features, namely [Go 1.23](#).

Objectives

- Develop an application in Go, respecting idioms and nomenclatures
- Master the concepts of concurrency and parallel programming
- Structuring, testing and debugging your microservices applications

Target audience

Project managers and developers.

Prerequisites

- Knowledge of a programming language
- [Test My Knowledge](#)

Technical requirements

- Have an IDE with the Golang plugin installed
- Have the latest version of Go installed

Our Go training program

Introduction to Go

- Introduction and history of Go
- The strengths (and weaknesses) of language
- Memory and Garbage Collector
- Competition and scalability
- Error handling
- Code learning and writing speed
- Compilation and execution speed
- Environment setup and configuration
- Go tools (formatting, compilation, etc.)
 - go fmt
 - go build
 - go test
 - go get
 - go mod
 - and the others...
- The basics
 - Syntax and reading direction
 - Basic types
 - Variables (and their assignment)
 - Constants
 - Packages and imports
 - Pointers
 - Collections (arrays, slices, maps)
 - Functions (definition, returns and accessibility)
 - Conditional loops and branches

Advanced programming

- Complex types
 - The structures
 - The composition
 - The functions
 - Assertion and type conversion
 - Interfaces and polymorphism
 - Errors

- Storage and exchange formats
 - Reading and writing to disk
 - Accessing and using a database (PostgreSQL)
 - ORM in Go.
 - XML encoding / decoding
 - JSON encoding/decoding
- Competition
 - What's the point of competition?
 - Parallelism and competition
 - Atomic functions
 - Go routines
 - Mutex
 - The channels
 - Other synchronization tools
- The tests
 - Creating a unit test
 - Launching unit tests
 - Code coverage
 - Malformed code detection
 - Breed detection conditions

HTTP server and REST API (microservices)

- HTTP server
 - Creating an HTTP server in Go
 - First handler (net/http) for server calls
 - Error handling and HTTP return codes
 - The different mux
- Introduction to REST and API creation
 - First API Rest service
 - JSON object mapping management
 - Adding services to the Rest API
 - API documentation: Swagger
- Testing
 - Testing API endpoints
 - Benchmark API performance

Complementary Module (1 day) - Creating your web API

- Creating a server
- Gin installation
- Route creation
- CRUD webservice generation
- HTML template
- Composability
- Variables and logic
- Connecting to Postgres
- Does ORM exist in Go?
- Creation of a
- Calling up ReactJS and installing a React client
- Calling up a webservice
- Customer display
- Creating microservices

- Creating Dockerfiles and Dockercompose
- MVC pattern

Companies concerned

This training course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the training to come, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, with brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.