# Formation Go: les fondamentaux du langage

*Développez backends et web services en Go*

N° ACTIVITÉ : 11 92 18558 92 | TÉLÉPHONE : 01 85 77 07 07 | E-MAIL : inscription@hubformation.com

Golang est en train de devenir le langage de prédilection du cloud. Lancé par Google en 2012, il est facile à installer et riche en bibliothèques dans de nombreux domaines. Statiquement typé, succinct, muni d'un ramasse-miette performant, et doté d'un modèle de concurrence limpide, il est rapide à compiler, peu gourmand en mémoire et permet d'utiliser pleinement tous les ressources de vos machines.
Ces caractéristiques en font le langage de choix pour l'implémentation de microservices modernes, simples et légers.
A l'issu de cette formation vous maîtriserez les aspects fondamentaux de la programmation en Go. Vous serez à même d'utiliser l'écosystème Go pour créer rapidement des applications performantes et facilement maintenables, à intégrer à votre système d'information. Vous retrouverez le bonheur de coder côté backend et partirez confiants dans la réalisation de vos objectifs.

| | |
|---|---|
| Référence | GOFND |
| Durée | 3 jours (21h) |
| Tarif | à partir de 2 100 €HT |

## PROCHAINES SESSIONS

Pour connaître les prochaines dates ou organiser un intra-entreprise, contactez-nous, nous vous répondrons sous 72 heures.

## Objectifs

| Maîtriser les fondamentaux du langage Go
| Définir une interface en ligne de commande
| Produire des logs
| Mettre à disposition des services HTTP
| Accéder à une base de donnée avec Go
| Empaqueter et déployer une application Go dans un conteneur docker

## Public

| Développeurs back ou full stack.

## Prérequis

| Etre à l'aise avec les concepts généraux de la programmation, des standards du web et des bases de données.
| Savoir cloner un repository git
| Savoir lancer des commande shell

## Programme de la formation

**Module 1: Introduction**
| Présentations, organisation des journées
| Go: Concepts-clés, contexte et historique

**Module 2: Fondamentaux et classiques**
| Compilation, exécution
| Packages, imports
| Types, variables, constantes
| Fonctions
| Structures conditionnelles
| Itérations
| Ateliers: Exercice autour des fonctions et des structures conditionnelles
| Exercice sur les itérations

**Module 3: Structures et pointeurs**
| Defer/panic/recover
| Structures
| Pointeurs
| Atelier: Jeu d'agilité mentale en rapport avec les pointeurs et les structures

**Module 4: Types agrégateurs fondamentaux**
| Tableaux
| Slices
| Maps
| Itérations sur ces types
| Closures
| Ateliers: Exercice de manipulations des tableaux et slices
| Exercice sur les closures

**Module 5: Méthodes et interfaces**
| Méthodes, récepteurs
| Interfaces & Duck Typing
| L'interface vide
| Assertions de type et type switches
| Ateliers: - Exercice sur les méthodes à recepteur de type struct - Exercice sur les type switches et recover

**Module 6: Programmation concurrente**
| Goroutines
| Channels
| Itérations sur channels
| Lecture/écriture (non/) bloquantes sur les channels
| Select : multiplexage
| Ateliers: Exercice sur l'utilisation des goroutines
| Exercice sur l'utilisation des channels pour la synchronisation des goroutines

**Module 7: tooling**
| Vérification des installations : git, docker
| Clonage du repository git et préparation du workspace
| Le tooling Go (run, build, test, ...)
| Le Makefile
| Les vendors
| Présentation du projet à coder

**Module 8: CLI : La ligne de commande**
| Lancer un programme en ligne de commande
| Natif et bibliothèques
| Le parsing des arguments
| Atelier: Compléter la gestion CLI de l'application du projet

**Module 9: Logging**
| Logger son application
| Natif et bibliothèques
| Configurer son logger
| L'exploitation des logs
| Atelier: Mise en place des logs dans l'appli

**Module 10: Concurrence, multi-threading, tests unitaires**
| Les solutions de Go (routines, channel)
| Multiplexage de channel (select)
| Les tests
| Atelier: Mise en oeuvre avec un compteur de requêtes

**Module 11: Accès aux données**
| Architecture des couches d'accès aux données
| Description du modèle de données
| Interface, DAO et Factory
| Les tests avec Mock
| Ateliers:Compléter la couche "modèle" de l'application du projet

| Compléter la factory, l'implémentation Mock
| Implementations MongoDB ou PostgreSQL

**Module 12: Service Web**
| Les package web de Go : natif et bibliothèques
| Architecture en couche du web
| Les middlewares
| Routage et Endpoints
| Les tests Web
| Atelier: Compléter le package "web" de l'application du projet

**Module 13: Build et déploiement**
| Dockerfile d'un programme Go
| Lancer son programme et sa base de données (compose)
| Tester son service

## Méthode pédagogique

Chaque participant travaille sur un poste informatique qui lui est dédié. Un support de cours lui est remis soit en début soit en fin de cours. La théorie est complétée par des cas pratiques ou exercices corrigés et discutés avec le formateur. Le formateur projette une présentation pour animer la formation et reste disponible pour répondre à toutes les questions.

## Méthode d'évaluation

Tout au long de la formation, les exercices et mises en situation permettent de valider et contrôler les acquis du stagiaire. En fin de formation, le stagiaire complète un QCM d'auto-évaluation.

---

## Accessibilité

Les sessions de formation se déroulent sur des sites différents selon les villes ou les dates, merci de nous contacter pour vérifier l'accessibilité aux personnes à mobilité réduite.
Pour tout besoin spécifique (vue, audition...), veuillez nous contacter au 01 85 77 07 07.