

Développement Go: Initialiser un projet

Stéphane Karraz
ESGI - cours électif de Go - 2022

Création d'un projet

En Go, nous devons créer un module pour pouvoir travailler et traquer nativement les dépendances, donc d'autres modules, importées et utilisées.

Pour initialiser un projet:

```
$ mkdir monnouveauprojet && cd monnouveauprojet  
$ go mod init mesprojets/monnouveauprojet
```

Un fichier go.mod devrait être créé, contenant la déclaration du module, la version du compilateur, ainsi que les futurs dépendances externes.

Point d'entrée

Créez un fichier main.go comme ceci:

```
package main
```

```
import "fmt"
```

```
func main() {  
    fmt.Println("hello world")  
}
```

Exécuter le code

Le Go est un langage compilé, mais vous pouvez demander au compilateur de l'exécuter directement:

```
$ go run main.go
```

```
> hello world
```

Compiler son code

On peut générer un exécutable comme ceci:

```
$ go build
```

Un exécutable du nom de votre projet doit exister dans votre dossier, vous pouvez le lancer classiquement:

```
$ ./monprojet
```

```
> "hello world"
```

Ajouter une dépendance

Le client permet d'en ajouter avec la commande “go get”, par exemple:

```
$ go get -u github.com/commander-cli/cmd@v1.0.0
```

Le paramètre “-u” permet d’indiquer qu’on souhaite mettre à jour le fichier go.mod de notre projet.

Cela génère aussi un fichier go.sum, listant les hashes des versions de nos modules externes installés.

\$GOPATH

Avant cette gestion en module, on utilisait une variable d'environnement appelée GOPATH pour indiquer le répertoire de travail.

Vous trouverez sans doute encore des tuto le mentionnant ainsi que des astuces sur sa gestion.

Le langage a bien évolué depuis, et cette manière de travailler est dépréciée et de plus en plus mise de côté au fil des versions.