



Workshop GO

B-INN-000

Day 02

Mini Projets

Task 00

Infos utiles



Quelques liens utiles :

- <https://gobyexample.com/>
- <https://golang.org/pkg/>
- <https://gist.github.com/leg0ffant/3bee4829ce2ad8fd026c>
- <https://golang.org/ref/spec>
- <https://talks.golang.org/2015/gophercon-goevolution.slide#1>

Projet 1

Mouli-norme

Dans ce projet, vous devrez coder une mini mouli-norme, vous aurez l'occasion de vous familiariser avec les tableaux, les goroutines et les appels système (ex : read, open, close ...).

Votre mouli-norme devra être capable d'ouvrir un fichier, de le lire et de détecter un maximum d'erreur de norme que possible, comme, par exemple, une ligne trop longue, ou un espace après un retour à la ligne.

Si vous souhaitez toucher aux goroutines, vous pouvez créer une goroutines pour chaque vérification (une goroutine pour la vérification de la taille des lignes, une autre pour les retours à la ligne ...).

Vous êtes libre de choisir si votre programme prend le fichier en paramètre, ou si votre programme vérifiera tous les fichiers dans le dossier actuel...



Packages utiles :

<https://golang.org/pkg/os/>

<https://golang.org/pkg/io/ioutil/>

<https://golang.org/pkg/fmt/#Printf>

<https://golangbot.com/goroutines/>

<https://gobyexample.com/goroutines>

Exemple d'output :

```
[pflorent@localhost moulinorme_go_2018]$ ./check_coding_style
  main.c Space after a ';' - line : 14 - col : 8 -> [ERROR]
  main.c Too long line - line : 13      -> [ERROR]
[number of errors] 2
```

Projet 2

Mini chat en réseau

Dans ce projet, vous devrez réaliser 2 programmes, une partie client et une partie serveur. Vous devrez réaliser un chat en réseau (tcp).

Pour cela vous aurez besoin de bonnes bases sur les goroutines et de quelques notions de réseaux (faible). Vous serez amené à travailler avec des nouvelles libs tel que Net ou io, n'hésitez pas à bien lire les pages en question.

Concernant la gestion des utilisateurs, vous ~~pouvez~~ devez gérer chaque nouvelle connexion dans une goroutine.



Packages utiles :

- <https://golang.org/pkg/net/>
- <https://golang.org/pkg/net/#Dial>
- <https://golang.org/pkg/net/#Listen>
- <https://golang.org/pkg/os/>
- os.Stdin

```
go run client.go
go run client.go 80x10
JOK3R@epitech.eu> ~/projets/G0/Day02 go run client.go
Enter a pseudo: aimbot
Enter text:
17:09 aimbot s'est connecté
test
17:09 aimbot : test
17:09 florent s'est connecté
17:09 florent : test
[]

go run server.go 80x11
JOK3R@epitech.eu> ~/projets/G0/Day02 go run server.go
Le serveur écoute à l'adresse ...:8754
aimbot s'est connecté
aimbot : test
florent s'est connecté
florent : test
[]

go run client.go
go run client.go 80x8
JOK3R@epitech.eu> ~/projets/G0/Day02 go run client.go
Enter a pseudo: florent
Enter text:
17:09 florent s'est connecté
test
17:09 florent : test
[]
```