DEV2 : tester automatiquement ses programmes

loig.jezequel@univ-nantes.fr

Objectifs du cours

Tester ses programmes

Quand on programme, on doit tester régulièrement son code, idéalement à chaque modification de celui-ci. Ça peut être laborieux.

Automatiser les tests

Pour ne pas avoir à sans cesse refaire des tests à la main on peut écrire et mettre à jour un jeu de tests qui pourra être automatiquement utilisé sur notre programme.

Évaluer l'efficacité du code

Un jeu de test permettra aussi de mesurer le temps de calcul de notre code sur un large ensemble de cas, ce qui permet d'évaluer son efficacité.

Automatisation des tests en Go

La base pour tester les paquets d'un module

- La commande *go test*
- Le paquet *testing*

La commande *go test*

- Lit les fichiers se terminant par _test.go du dossier courant,
- appelle une à une toutes les fonctions de test dans ces fichiers,
- affiche le résultat de ces tests (réussi ou non).

Fonction de test

- Son nom commence par Test,
- a un unique argument t de type *testing.T,
- appelle la fonction t.Fail() quand un test est raté.

Évaluation de l'efficacité du code

La base pour évaluer l'efficacité des paquets d'un module

- La commande *go test -bench=.*
- Le paquet *testing*

La commande *go test -bench=.*

- Lit les fichiers se terminant par _test.go du dossier courant,
- appelle toutes les fonctions de benchmarking dans ces fichiers,
- affiche le résultat de ces évaluations.

Fonction de benchmarking

- Son nom commence par Benchmark,
- a un unique argument b de type *testing.B,
- doit utiliser b.N fois le code à évaluer.

Quelques précisions sur *go test*

Paquets à tester

On peut les préciser en listant leurs noms longs à la suite de la commande. Sinon, la commande travaille uniquement sur le dossier courant.

Tests et benchmarks

Par défaut, quand on fait l'évaluation de l'efficacité du code (*go test -bench=.*) cela exécute aussi les tests. Pour juste faire l'évaluation on peut préciser la commande :

go test -run=Bench -bench=.

-run= et -bench=

Ces arguments permettent en fait de filtrer les tests à réaliser (run) et les benchmarks à réaliser (bench), on peut mettre n'importe quelle expression rationnelle après le =

Quelques ressources

- Infos sur la commande *go test* : *go help test*
- Documentation du paquet testing : `https://pkg.go.dev/testing`