

Le langage Go

Go et le web

Serveur HTTP

Qui dit faire du web dit

Un Client et un Serveur

Go propose un serveur web natif

Exemple

```
1 package main
2
3 import (
4     "fmt"
5     "log"
6     "net/http"
7 )
8
9 // Cette fonction doit respecter cette signature pour être une HandlerFunc
10 func hello(w http.ResponseWriter, r *http.Request) {
11     // Ecriture d'un message en réponse HTTP
12     fmt.Fprintf(w, "Hello gophers!")
13 }
14
15 func main() {
16     // Associe un chemin à une fonction
17     http.HandleFunc("/", hello)
18
19     // Création du serveur HTTP écoutant sur le port 8080
20     if err := http.ListenAndServe(":8080", nil); err != nil {
21         log.Fatal(err)
22     }
23 }
```

**Go va exécuter une goroutine pour
chaque requête HTTP reçue**

C'est du natif et pourtant...

Les performances sont excellentes

Comparatif de performance

On peut aussi bien manipuler la requête que la réponse

```
// w <= Accéder à la réponse  
// r <= Accéder à la requête  
func hello(w http.ResponseWriter, r *http.Request) {  
}
```

Allons voir cela en code