Static server

GitHub repo

Dans certains cas, on souhaite juste héberger un site statique.

On pourrait se tourner vers apache ou nginx mais ce n'est pas ce que nous recherchons □□

Il est possible assez facilement grâce à **Gin** de rendre accessible notre site statique.

Pour cette démonstration, je possède l'architecture suivante :

```
static/
|  assets/
|  |  ...
|  index.html
|  script.js
|  ...
main.go
go.mod
go.sum
```

Mon site dans le dossier `static` est une application Angular (avec Angular router pour l'exemple haha) compilé en version de production.

Pas besoin d'aller très loin, notre fichier `main.go` ressemblera à ça :

```
package main

import "github.com/gin-gonic/gin"

func main() {
□router := gin.Default()

□router.Static("/", "./static")

□router.Run(":8080")
}
```

Pas besoin de détailler, les fonctions parlent d'elles même.

Si je me rends sur mon site, on voit que tout va BIEN :

Untitled

Je me rends sur une autre page de mon site en lançant un combat, et là aussi tout va BIEN :

Untitled

> SAUF QUE !
> Si je décide de rafraîchir ma page, avec cet URL là, et bien j'obtiens une belle 404...

Untitled

Cela vient du fait que le router va chercher bêtement un fichier au chemin `/fight/charizard/blastoise` dans notre dossier `static` alors qu'il devrait passer ce chemin à notre application Angular, c'est un problème récurent avec les applications web.

Il existe heureusement une solution, il suffit de dire à **Gin** que s'il ne trouve pas le chemin en question dans l'arborescence de dossier, il doit alors interroger l'application Angular, qui se chargera elle-même de renvoyer une erreur 404 si le chemin n'existe effectivement pas.

```
package main

import "github.com/gin-gonic/gin"

func main() {
	router := gin.Default()

	router.Static("/", "./static")

	router.NoRoute(func(c *gin.Context) {
		c.File("./static/index.html")
	})

	router.Run(":8080")
}
```

Et là, si on rafraichit, on retrouve notre beau combat dans notre arène

---

Revision #1
Created 2022-05-09 19:30:51 UTC by Noé Larrieu-Lacoste
Updated 2022-05-09 19:31:43 UTC by Noé Larrieu-Lacoste