

Live reload

Recompiler notre code à chaque fois que l'on change notre code, arrêter le serveur et le relancer...

Tout ça est long et fastidieux ! Surtout pendant le développement !

En nodeJS certains se souviendront de nodemon qui permettait de surveiller les changements dans nos fichiers, et les recompiler à la volée.

En Go il existe différents outils permettant de faire cela, nous allons utiliser [Air](#)

Air installation

Pour installer Air en tant qu'exécutable reconnu par notre machine, il suffit de faire la commande suivante :

```
go install github.com/cosmtrek/air@latest
```

Ensuite, nous allons "pimper" un peu la configuration de cet outil pour avoir un peu de couleur ☺.

Pour ce faire, nous allons créer le fichier `.air.conf` à la racine de notre projet et le remplir ainsi :

```
# .air.conf
# Config file for [Air](https://github.com/cosmtrek/air) in TOML format

# Working directory
# . or absolute path, please note that the directories following must be under root.
root = "."
tmp_dir = "tmp"

[build]
# Just plain old shell command. You could use `make` as well.
cmd = "go build -o ./tmp/main ." # replace by main.exe if on windows !
# Binary file yields from `cmd`.
bin = "tmp/main" # replace by main.exe if on windows !
# Customize binary.
# Watch these filename extensions.
include_ext = ["go", "tpl", "tmpl", "html"]
```

```
# Ignore these filename extensions or directories.
exclude_dir = ["assets", "tmp", "vendor", "frontend/node_modules"]
# Watch these directories if you specified.
include_dir = []
# Exclude files.
exclude_file = []
# It's not necessary to trigger build each time file changes if it's too frequent.
delay = 1000 # ms
# Stop to run old binary when build errors occur.
stop_on_error = true
# This log file places in your tmp_dir.
log = "air_errors.log"

[log]
# Show log time
time = false

[color]
# Customize each part's color. If no color found, use the raw app log.
main = "magenta"
watcher = "cyan"
build = "yellow"
runner = "green"

[misc]
# Delete tmp directory on exit
clean_on_exit = true
```

Attention à la ligne 11 et 13 !

Désormais, il suffit de lancer la commande `air` à la racine du projet pour le lancer et surveillez les changements de code :

Untitled

Au moindre changement, on pourra voir que le script le détecte et recompile aussitôt

Untitled

Revision #1

Created 9 May 2022 19:26:38 by Noé Larrieu-Lacoste

Updated 9 May 2022 19:27:34 by Noé Larrieu-Lacoste