GO

Live Reload in Go with Air

Author
Gurleen Sethi on 29 May 2022

TheDeveloperCafe

Getting Started

In this article you will learn how to live reload go applications with one of the popular packages out there air. First we will learn how to setup up air and start using it right away, then we will see how to customize air with `.air.toml` file.

Installing air

There are many ways to install **air** as seen in the installation section of air's documentation. The way I am going to use here is via `go install` (for `go install` to work you would `H/bin` in your shell path).

Table of Contents

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

Live reload go code

In a new directory create a `main.go` file with the following content:

main.go

```
package main

import "fmt"

func main() {
    fmt.Println("Air")
}
```

and run the following command to see live reloading in action:

```
air
```

You should see the following output:

```
watching .
!exclude tmp
building...
running...
Air
```

Now go ahead change the code and save the file:

```
package main

import "fmt"

func main() {
    fmt.Println("Air Cha                  essage
}
```

≡ Table of Contents

Air will detect that the file is changed and you will see the new message:

```
watching .
!exclude tmp
building...
running...
Air Changed
```

Air creates a directory called `tmp` where it stores the built binary and executes it.

Configuring air with config file

You can cutomize **Air** using a `.air.toml` file.

Run the following command in the project directory to generate a `.air.toml` file.

```
air init
```

This will generate a `.air.toml` file with the following output:

.air.toml

```
root = "."
testdata_dir = "testdata"
tmp_dir = "tmp"

[build]
  bin = "./tmp/main"
  cmd = "go build -o ./tmp/main ."
  delay = 1000
  exclude_dir = ["assets", "tmp", "vendor", "testdata"]
  exclude_file = []
  exclude_regex = ["_test.go"]
  exclude_unchanged = false
  follow_symlink = false
  full_bin = ""
  include_dir = []
  include_ext = ["go", "tpl", "tmpl", "html"]
```

☰ Table of Contents

```
  kill_delay = "0s"
  log = "build-errors.log"
  send_interrupt = false
  stop_on_error = true

[color]
  app = ""
  build = "yellow"
  main = "magenta"
  runner = "green"
  watcher = "cyan"

[log]
  time = false

[misc]
  clean_on_exit = false

[screen]
  clear_on_rebuild = false
```

Let's take an example, set `clean_on_exit` to `true`, this will delete the `tmp` (air stores the build binary file) folder when you exit **air**.

```
[misc]
  clean_on_exit = true
```

Run the **air** command by passing the config using `-c` option:

```
air -c .air.toml
```

To find the documentation of all the properties of `.air.toml` read the air_example.toml file.

Thank you for reading this article 🙏🏼 I hope it was helpful.

≡ Table of Contents

Get updates via email 👉

Get notified once/twice per month when new articles are published.

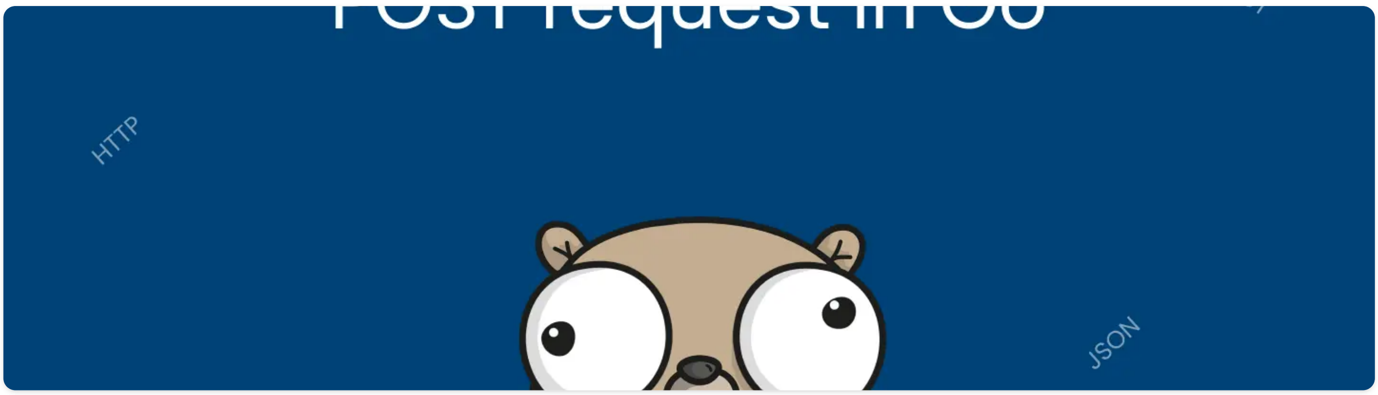
Your email

Subscribe

Related Articles

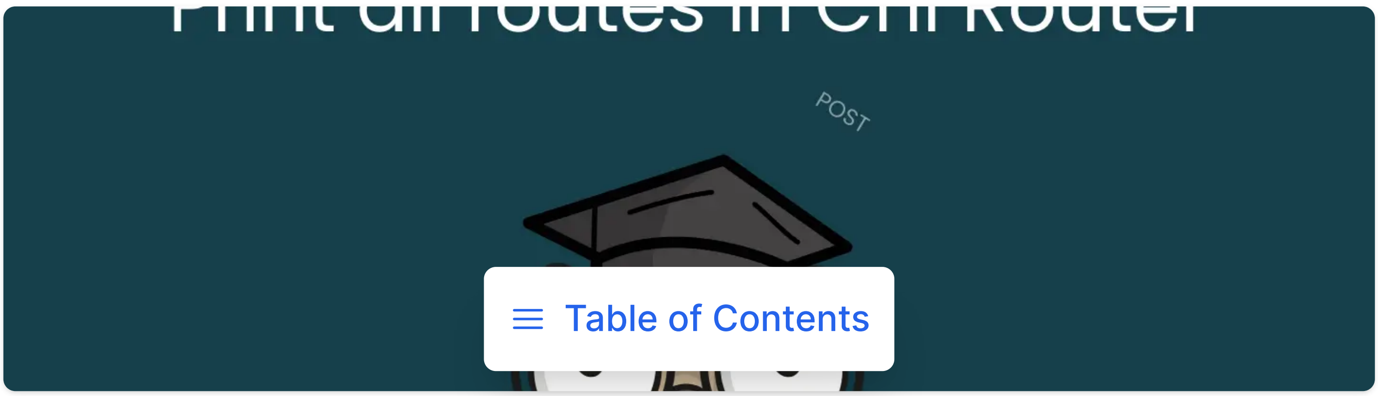
GO

Logging in Go with slog



GO

Make POST request in Go using net/http



≡ Table of Contents

GO

Print all routes in Chi router

TheDeveloperCafe

☰ Table of Contents