

Building APIs using Go

Elton Minetto

@eminetto

<http://eltonminetto.net>

This is my approach

based on other projects...

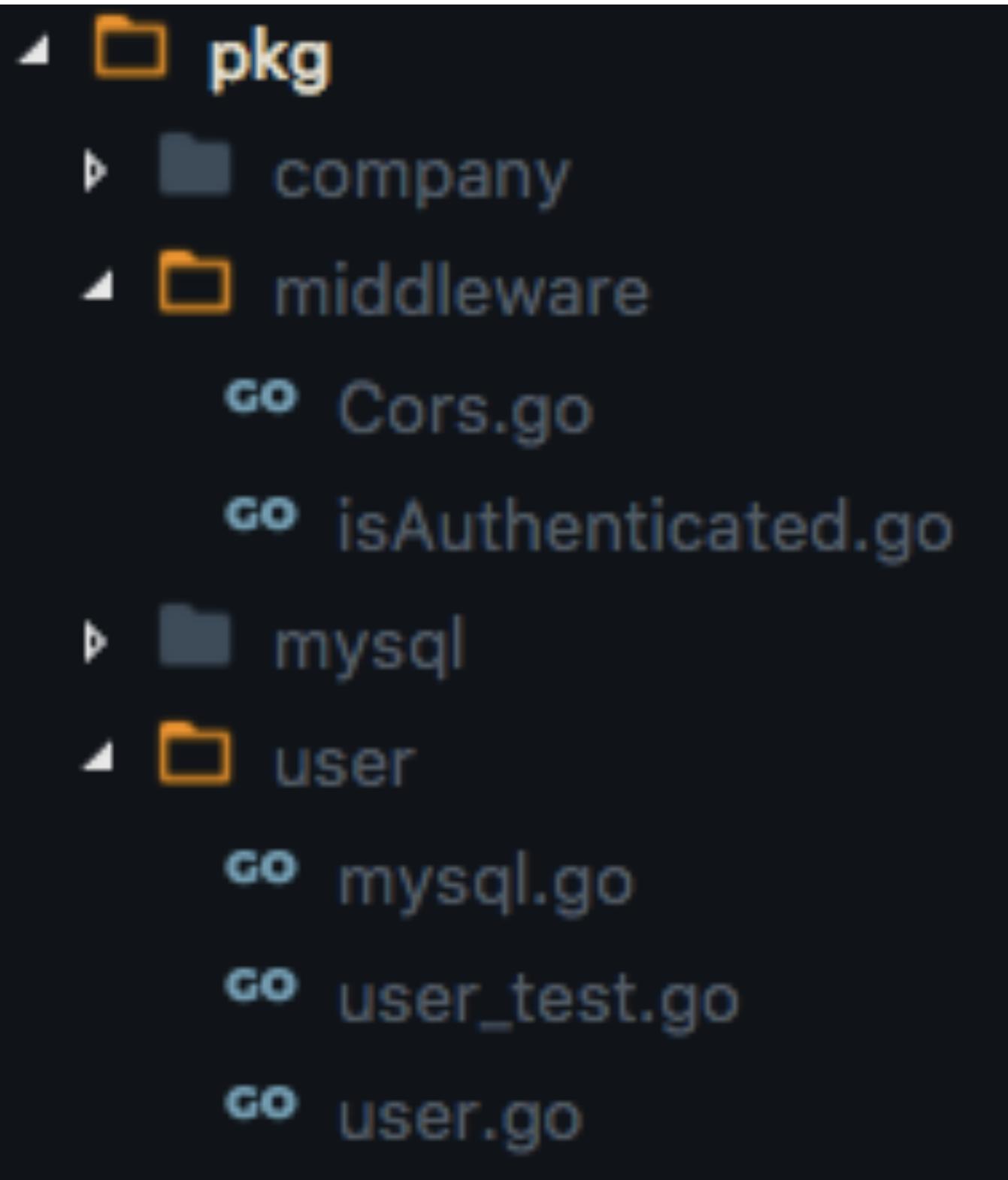
No frameworks!

Stdlib + small pieces

- github.com/asaskevich/gvalidator
- github.com/codegangsta/negroni
- github.com/go-sql-driver/mysql
- github.com/gorilla/mux
- github.com/joho/godotenv
- gopkg.in/gorp.v1
- github.com/rubenv/sql-migrate

Project structure

```
api
  handler
    company.go
    user.go
    main.go
bin
  * api
  * cli
cli
  go.test.sh
cmd
  main.go
config
  dbconfig.yml
  * dev.env
  * production.env
  * testing.env
doc
migrations
pkg
  company
  middleware
  mysql
  user
  vendor
    .gitignore
    coverage.txt
    docker-compose.yml
    Gopkg.lock
    Gopkg.toml
  Makefile
README.md
```

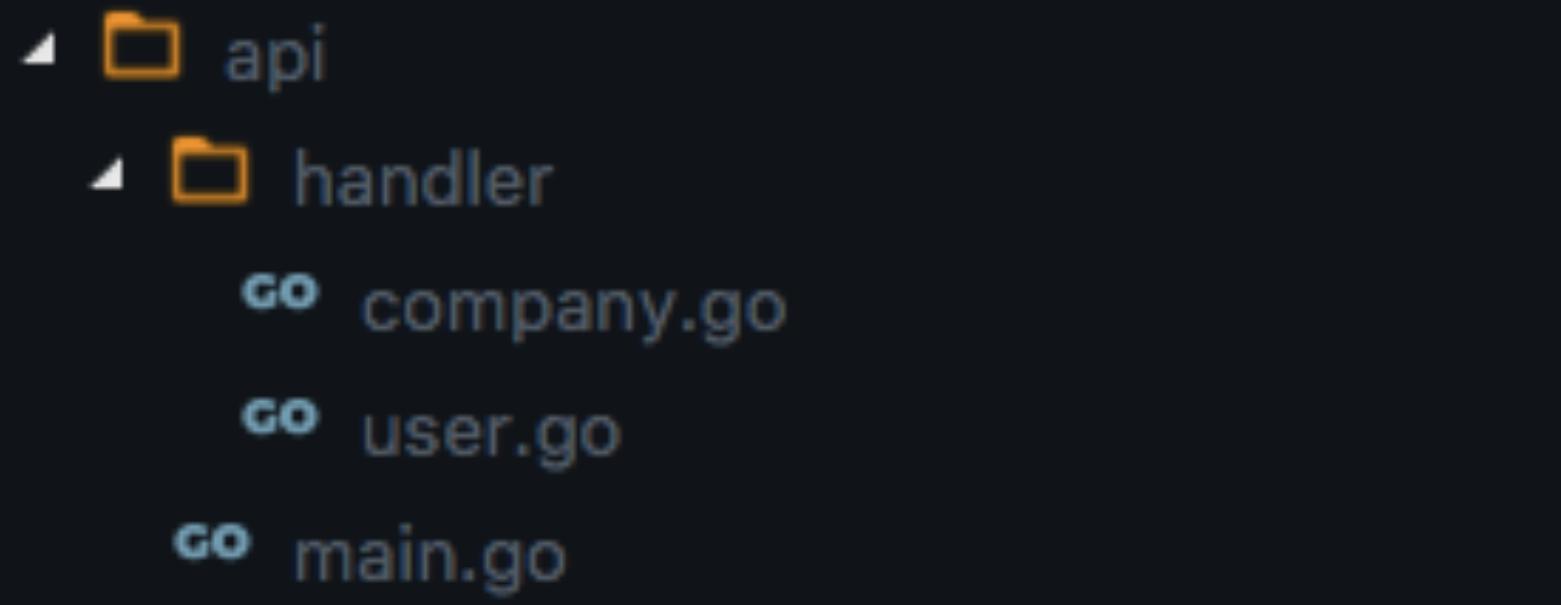


pkg

```
user/
  user.go (structs and service interface)
  user_test.go (tests)
  mysql.go (implementation of the service interface)

middleware/
  Cors.go -> add cors headers
  isAuthenticated.go -> request authentication
```

api



handler/
 user.go (define routes, handler implementation)

main.go (act as a dependency injection container)

```
└─ cmd
    └─ main.go
```

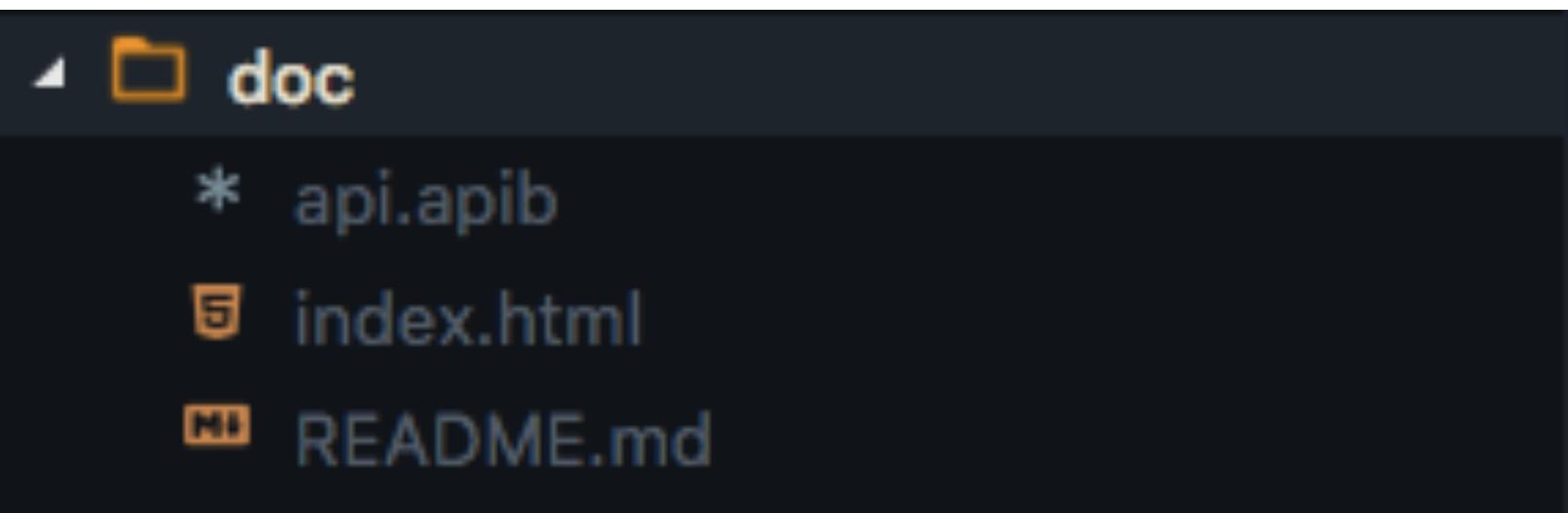
cmd

```
└─ bin
  * api
  * cli
```

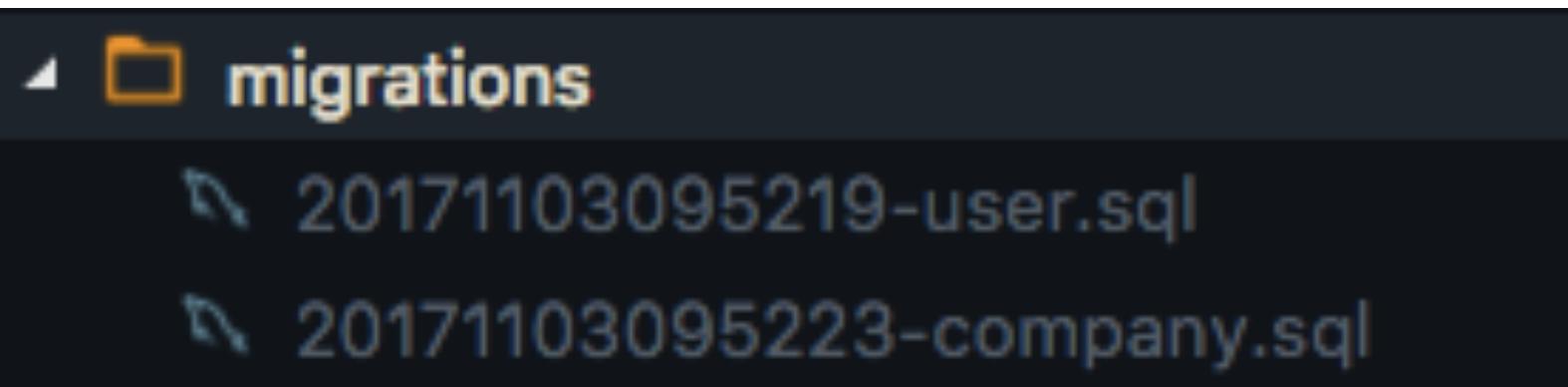
bin

```
▲ config
  ▼ dbconfig.yml
  * dev.env
  * production.env
  * testing.env
```

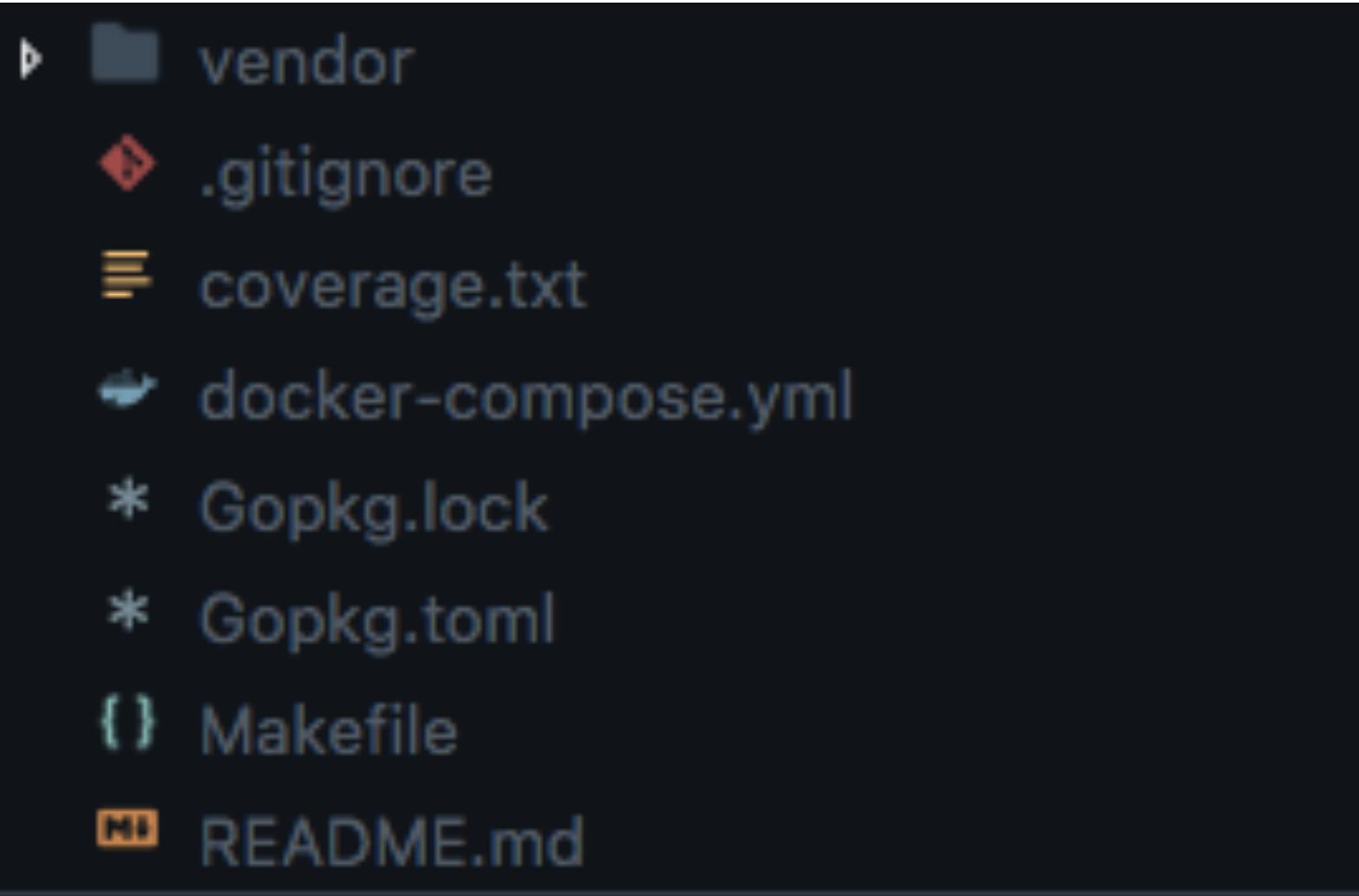
config



doc



migrations



**vendor (e other
files)**

Links

<https://github.com/eminetto/go-web-skel>

Desenvolvendo APIs em Go usando Middlewares

Definindo APIs com o API Blueprint

Gerenciando dependências em Golang

Contato

<http://twitter.com/eminetto>

<http://eltonminetto.net>

<http://asemanago.com.br> (+550 assinantes)

eminetto@gmail.com