Hexagonal

Pallat Anchaleechamaikorn

> Technical Coach
> Infinitas by KrungThai
> Arise by Infinitas

yod.pallat@gmail.com
https://github.com/pallat
https://dev.to/pallat

https://go.dev/tour (Thai)
https://github.com/uber-go/guide (Thai)

Facebook: Pallat

https://www.facebook.com/yod.pallat

Facebook: GoGetTH

https://www.facebook.com/gogetth

dev.to/pallat

https://dev.to/pallat

Exercise

ข้อใดทำไม่ได้

1. `var s []string = nil`
2. `var s []string =[]string(nil)`
3. `var s = nil`
4. `var s []string = []string{}`

Zero Value ของ type ใดไม่เท่ากับ nil

1. `[]struct`
2. `[0]func()`
3. `*bool`
4. `any`

s = ?

```
a := [...]int{-1, 0, 1, 2, 3, 4, 5, 6}
s := a[0:5]
```

Hello World

```go
package main

import "fmt"

func main() {
    fmt.Println("Hello World")
}
```

Add(int, int)

$$sum = a + b$$

example: $1 + 2 = 3$

```
func(a, b int) int
```

Add(float64, float64)

$$sum = a + b$$

example: $1 + 2 = 3$

```
func(a, b float64) float64
```

Add(T, T)

$sum = a + b$

example: $1 + 2 = 3$

```
var a, b int = 1, 2
var y, z float64 = 1, 2

Add(a, b)
Add(y, z)
```

Quick Start

https://pkg.go.dev/net/http

First API using net/http

```go
package main

import (
    "io"
    "log"
    "net/http"
)

func main() {
 // Hello world, the web server

 helloHandler := func(w http.ResponseWriter, req *http.Request) {
    io.WriteString(w, "Hello, world!\n")
 }

    http.HandleFunc("/hello", helloHandler)
    log.Fatal(http.ListenAndServe(":8080", nil))
}
```

Web Framework Benchmark

https://www.techempower.com/benchmarks/#section=data-r19&hw=ph&test=plaintext

https://github.com/smallnest/go-web-framework-benchmark

GoFiber

https://docs.gofiber.io/

https://github.com/gofiber/fiber

GoFiber: Example

```go
package main

import "github.com/gofiber/fiber/v2"

func main() {
    app := fiber.New()

    app.Get("/", func(c *fiber.Ctx) error {
        return c.SendString("Hello, World 👋!")
    })

    app.Listen(":3000")
}
```

GoFiber.io

★ Some values returned from *fiber.Ctx
are not immutable by default

Issue

https://github.com/gofiber/fiber/issues/426

อย่าเพิ่งใช้ fiber ถ้ายังไม่ได้อ่าน doc

https://dev.to/pallat/yaaephingaich-fiber-thaayangaimaidaan-doc-2gnh

Go Tutorial

https://go.dev/doc/tutorial/

https://go.dev/doc/tutorial/web-service-gin

Gin-Gonic

https://github.com/gin-gonic/gin
https://gin-gonic.com/

Gin-Gonic Example

```go
package main

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

func main() {
    r := gin.Default()
    r.GET("/ping", func(c *gin.Context) {
        c.JSON(200, gin.H{
            "message": "pong",
        })
    })
    r.Run() // listen and serve on 0.0.0.0:8080 (for windows "localhost:8080")
}
```

Exercise: Ping-Pong Handler

Move the anonymous handler function to a named function.

Requirement

Todo App store in memory

> **POST** */todos*
>
> **GET** */todos*
>
> **PATCH** */todos*
>
> **DELETE** */todos/:**id***

Todo Package

> github.com/gophernment/api/todo

```
mkdir todo
```

Todo Model

```
type Todo struct {
    ID        uint      `json:"id"`
    Title     string    `json:"title"`
    Done      bool      `json:"done"`
    CreatedAt time.Time `json:"created_at"`
    UpdatedAt time.Time `json:"updated_at"`
    DeteledAt time.Time `json:"deteled_at"`
}
```

Todo Handler

gin.HandlerFunc

```
type HandlerFunc func(*Context)
```

Toto Handler

```
func List(c *gin.Context) {
        c.JSON(200, todos)
}
```

Gracefully shutting down

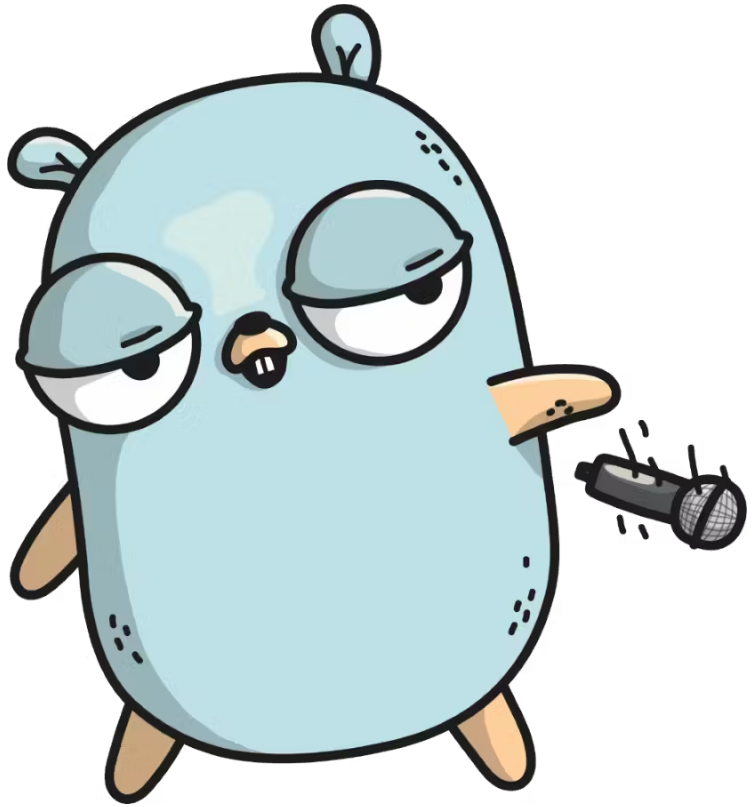

image created by *Pallat Anchaleechamaikorn* **© 2023**

ListenAndServe

The Kubernetes termination lifecycle

https://cloud.google.com/blog/products/containers-kubernetes/kubernetes-best-practices-terminating-with-grace

> In practice, this means your application needs to handle the SIGTERM message and begin shutting down when it receives it. This means saving all data that needs to be saved, closing down network connections, finishing any work that is left, and other similar tasks.

graceful example

```
ctx, stop := signal.NotifyContext(context.Background(), syscall.SIGINT, syscall.SIGTERM)
defer stop()

go func() {
    if err := server.ListenAndServe(); err != nil && err != http.ErrServerClosed {
        log.Fatalf(“listen: %s\n”, err)
    }
}()

<-ctx.Done()
stop()
fmt.Println("shutting down gracefully, press Ctrl+C again to force")

timeoutCtx, cancel := context.WithTimeout(context.Background(), 5*time.Second)
defer cancel()

if err := server.Shutdown(timeoutCtx); err != nil {
    fmt.Println(err)
}
```

test shutdown

```
sudo lsof -i :8080
kill -15 [PID] || kill -SIGINT [PID]
kill -SIGTERM [PID]
```

Configurations

https://12factor.net/

Viper

https://github.com/spf13/viper

.env auto loading

https://github.com/joho/godotenv

direnv

https://github.com/direnv/direnv

https://gist.github.com/rmtuckerphx/4ace28c1605300462340ffa7b7001c6d

Our Config

.env

```
ADDR=":8081"
```

Dockerfile

```
FROM cgr.dev/chainguard/go:latest as build
WORKDIR /app
COPY go.mod go.sum ./
RUN go mod download
COPY . ./
RUN go build -o api

## Deploy
FROM cgr.dev/chainguard/static:latest
COPY --from=build /app/api .
EXPOSE 8080
USER nonroot:nonroot
CMD ["/api"]
```

https://twitter.com/sidpalas/status/1637839918448754688?s=20

```
FROM cgr.dev/chainguard/go:latest as build

WORKDIR /app
COPY go.mod go.sum ./
RUN --mount=type=cache,target=/go/pkg/mod \
  --mount=type=cache,target=/root/.cache/go-build \
  go mod download

COPY . .
RUN go build \
    -ldflags "-linkmode external -extldflags -static" \
    -o api

FROM cgr.dev/chainguard/static:latest
LABEL version="X.Y.Z"
COPY --from=build /app/api .
EXPOSE 8080

CMD ["/api"]
```

Docker build

```
docker build -t todo:latest -f Dockerfile .
```

Docker run

```
docker run -p:8081:8081 --env-file ./.env --name mytodo todo:latest
```

add route GET /todos

```
r.GET("/todos", todo.List)
```

POST /todos

> **POST** */todos*

```
{
    "title": "go punching"
}
```

return status code 201

PATCH /todos

> **PATCH** */todos*

```
{
    "id": 1,
    "title": "go lifting"
}
```

return status code 200

DELETE /todos/:id

> **DELETE** */todos:**id***

return status code 204

database/sql

```
import (
    "database/sql"
    _ "github.com/mattn/go-sqlite3"
)
```

the Blank identifier

(underscore)

import with need the side effects

_ "github.com/mattn/go-sqlite3"

windows

_ modernc.org/sqlite

_ Blank identifier

```
import (
    _ "myproject/effect"
)
```

we don't want to use any stuff from that package

we need just the effect from `init()` in there

SQLite extension

https://marketplace.visualstudio.com/items?itemName=qwtel.sqlite-viewer

database/sql with sqlite

```go
func main() {
    db, err := sql.Open("sqlite", "./database.sqlite")
    if err != nil {
        log.Fatal(err)
    }
    defer db.Close()

    // TODO: logic
}
```

Connect to Database

> /play/database

https://github.com/mattn/go-sqlite3/blob/master/_example/simple/simple.go

Make Todo store in SQLite

```
ISO8601: YYYY-MM-DD HH:MM:SS.SSS
"2006-01-02 15:04:05.000"
```

Object-Relational Mapping (ORM)

gorm.io

xorm.io

ent (entgo.io)

gorp

etc.

https://github.com/avelino/awesome-go#orm

Template

https://github.com/pallat/gotemplate