

# GoFrame

Django/FastAPI/SpringpythonRESTful APIweb servergoweb serverbeego/iris/ginGolang: goframe, beego, irisgin GoFrame~

GoFrame | (xiaobinzhao.github.io)

markdown

- gitee: [https://gitee.com/xiaobin\\_zhao/my-gf.git](https://gitee.com/xiaobin_zhao/my-gf.git)
- github: <https://github.com/XiaobinZhao/my-gf.git>

S2E7:)

1.
  - a.
  - b.
  - c.
  - d.
  - e.
  - f.
  - g.
  - h. Hotfix
    - master
  - o owner
  - o 21.09.1masterowner
  - o gitlabmerge requestmerge request{}.{}\_{}san.zhang\_feature-local-image,local-image
  - o commitidTAPD
  - o release
  - o hotfixmastertaghotfixtagtag
2. gogofmtpythonPEP8Java
- 3.
4. GoFrame
5. Scrum/kanban
  - a.
  - b. PRDPRD
  - c.
  - d. PMO
  - Gitlab
  - JavaMavenNexusdockerpython
  - SaaS TowerTeam ambitionatlassianJira tapd
  - SaaS WikiatlassianConfluence
  - BugJira
  - SonarQube

goGoFramepythonFastAPI

1. Web MVCGoFrameAPIcontrollersservicedaoxxxReq/xxxResxxxxInput/xxxxOutput
2. ormGoFramegdb
3. GoFramegcache
4. .alembic
5. Web APIswaagerAPIRESTfulAPIAPIHATEOASCollectionmock

- 
- 
- o
- o
- o
- o
- o
- 
- o go
  - apt-get
  -
- o GoFrame
- o Go Frame
  - - api
  - - service
  - - dao
  - - model
- o
  - cmd
  - api
  - controller
  - model
  - service
  - dao
- o Go Frame
  - 1. MVC
  - 2. service controller
  - 3. service dao
- 
- o 1. gf-cli
- o 2.
- o 3. gf clidao/do/entity
- o 4.
  - 1. main
  - 2.
  - 3. middleware
  - 4. openAPIDoc (swagger)
  - 5. controller
  - 6. service
  - 7. inputModel outputModel
  - 8. dao/do /entity
  - 9. errCode
  - 10. I18N
  - 11.
    - serv
    - er
    - db
    - log
  - 12.
  - 13. token
  - 14.
    -
  - 
  - 15.
- ...

6. APISwaagerUIredoc
7. GoFramegvalidweb API
8. GoFramegcfg
9. GoFrameglog
10. /GoFramegcmd
11. GoFramegf
12. GoFramegerrorgcode.
13. GoFrame
14. I18NGoFramei18n
- 15.
- 16.

- a. http client
- b.
- c. md5/ses/des/sha1
- d.

GoFrame

- MQBrokerProxyRabbitMQRocketMQKafka
- APIKong<https://github.com/Kong/kong> Spring Cloud Gateway
- DB<https://github.com/fliike/kingshard> <https://github.com/Qihoo360/Atlas> MySQL ProxySQL
- ESESLucene
- nginx

Proxy

1. CI/CDCl/CDJenkinsSSH+
2. Prometheus/Zabbix+Grafana.
3. K8Sk8sAPIk8sRancherk8s - ([miaoyun.net.cn](http://miaoyun.net.cn))

~

GoFrame~

go

windowexelinuxgo1.16.13

**apt-get**

apt-get install golang

Debian9 golanggolang-1.7

1. <https://golang.google.cn/dl/wget> <https://golang.google.cn/dl/go1.16.13.linux-amd64.tar.gz>
2. mkdir -p /usr/local/lib
3. tar -xzf go1.16.13.linux-amd64.tar.gz -C /usr/local/lib
4. ~/.profile,

```
# golang
GOPATH=/root/go-workspace/
GOROOT=/usr/local/lib/go
export GO111MODULE=on
export GOPROXY=https://goproxy.io
export PATH=$GOPATH/bin:$GOROOT/bin:$PATH
```

- 5. source ~/.profile
- 6. go version

## GoFrame

Golang: goframe, beego, irisgin - GoFrame - GoFrame - PHP-Laravel, Java-SpringBootGo

- 1.
- 2.
- 3.
- 4.

Golang  
goframe  
Golang2019goframePASS  
goframe  
goframegoframegoframegoframe

## Go Frame

GoFrame

```
api
internal
  cmd
  consts
  controller
  model
  entity
  service
    internal
      dao
      do
manifest
resource
utility
go.mod
main.go
```

i18ntemplate

/		
api		/apiv1/apiv2...
internal		Golang internalinternalinternal
- cmd		
- consts		
- controller		//

- model		
- entity		
- service		
- dao		CURD
- do		dao
manifest		
- config		
- docker		Docker
- deploy		KubernetesYamlkustomize
resource		/
go.mod		Go Module
main.go		

## - api

api+controller

apiUI service

## - service

serviceBLL

## - dao

daoDAL

## - model

modelModel

modelentity/api/dao/service

[blocked URL](#)

[blocked URL](#)

## cmd

cmdserverserver

## api

serverapiReqReqReqReqcontroller

## controller

controllerReqserviceRes

## model

modelserviceInput/Outputmodel

## service

servicedaodaododaoEntityservice

## dao

daoORM

## Go Frame

### 1. MVC

GoFrameMVCMVC

### 2. servicecontroller

controllerReq/Res service

serviceInput/Output

controllerReqserviceserviceReservicecontrollerserviceReqservicecontroller

### 3. servicedao

daoservicedaodaoservicedaoservice

CURDdaodaodaodaoserviceservicedaoserviceservice+dao

:daoserviceservicedaoserviceserviceservicedao

gf - [GoFrame \(ZH\)-v2.0](#) - [GoFrame](#)

## 1. gf-cli

### 1. windows

<https://github.com/gogf/gf/releases>gf.exe%GOROOT%/bin%GOPATH%/bininstall

```
$ ./gf_windows_amd64.exe install
I found some installable paths for you(from $PATH):
Id | Writable | Installed | Path
0 | true | true | E:\go-workspace\bin
1 | true | false | D:\programs\go\bin
please choose one installation destination [default 0]:
gf binary is successfully installed to: E:\go-workspace\bin
```

```
$ ./gf_windows_amd64.exe install
I found some installable paths for you(from $PATH):
Id | Writable | Installed | Path
0 | true | true | E:\go-workspace\bin
1 | true | false | D:\programs\go\bin
please choose one installation destination [default 0]:
gf binary is successfully installed to: E:\go-workspace\bin
```

2.

```
git clone https://github.com/gogf/gf && cd gf/cmd/gf && go install
```

3.

```
gf -v
```

OK

```
GoFrame CLI Tool v2.0.6, https://goframe.org
GoFrame Version: v2.0.6 in current go.mod
CLI Installed At: E:\go-workspace\bin\gf.exe
CLI Built Detail:
  Go Version:  go1.17.7
  GF Version:  v2.0.6
  Git Commit:  2022-03-31 16:57:32
  66803fd6641e31c60f0489180668243ea092b87b
  Build Time:  2022-03-31 16:55:38
```

2.

SQLresource/doc

### 3. gf clidao/do/entity

1.

```
# GF-CLI
[gfcli]
# DAO(database)
[[gfcli.gen.dao]]
  link      = "mysql:root:mysql@tcp(192.168.212.117:3306)/myapp"
  descriptionTag = true
  noModelComment = true
```

2. gf cli

```
gf gen dao
```

```
> gf gen dao
generated: internal\service\internal\dao\desktop.go
generated: internal\service\internal\dao\internal\desktop.go
generated: internal\service\internal\dao\user.go
generated: internal\service\internal\dao\internal\user.go
generated: internal\service\internal\do\desktop.go
generated: internal\service\internal\do\user.go
generated: internal\model\entity\desktop.go
generated: internal\model\entity\user.go
done!
```

4.

### 1. main

main http server

```
gf cmdgf
```

```

func main() {
s := g.Server()
g.I18n().SetPath("resource/i18n") // i18ngresil18n
s.Group("/", func(group *ghttp.RouterGroup) {
    group.Middleware(
        service.Middleware().I18NMiddleware,
        service.Middleware().ResponseHandler,
    )
    //group.Bind(
    //    controller.User,    //
    //    controller.Desktop, //
    //)
    // pathmethodAPIstruct
    group.Group("/users", func(group *ghttp.RouterGroup) {
        group.GET("/", controller.User.List)
        group.POST("/", controller.User.Create)
        group.GET("/{uuid}", controller.User.Get)
        group.PATCH("/{uuid}", controller.User.Update)
        group.DELETE("/{uuid}", controller.User.Delete)
    })
    group.Group("/desktops", func(group *ghttp.RouterGroup) {
        group.GET("/", controller.Desktop.List)
        group.POST("/", controller.Desktop.Create)
        group.GET("/{uuid}", controller.Desktop.Get)
        group.PATCH("/{uuid}", controller.Desktop.Update)
        group.DELETE("/{uuid}", controller.Desktop.Delete)
    })
})
//
enhanceOpenAPIIDoc(s)
// Http Server
s.Run()
}

```

g.Server()ServerServerRun()Server80

gfweb

- WebServer
- Server
- Server
- Server
- https
- server

## 2.

gf//restful///map

gf+OpenAPIv3swagger

### 1. SwaggerUI

```

# HTTP Server.
[server]
    openapiPath    = "/api.json"
    swaggerPath    = "/swagger"

```

### 2.

main

```

        group.Bind(
            controller.User,    //
            controller.Desktop, //
        )

```

pathurl

```

        type UserUpdateReq struct {
g.Meta      `path:"/user" method:"put" summary:"" tags:"" `
        .....
        }

```

url

```

        s.Group("/", func(group *ghttp.RouterGroup) {
            group.Middleware(
                service.Middleware().I18NMiddleware,
                service.Middleware().ResponseHandler,
            )
            // pathmethodAPIstruct
            group.Group("/users", func(group *ghttp.RouterGroup) {
                group.GET("/", controller.User.List)
                group.POST("/", controller.User.Create)
                group.GET("/{uuid}", controller.User.Get)
                group.PATCH("/{uuid}", controller.User.Update)
                group.DELETE("/{uuid}", controller.User.Delete)
            })
        })

```

3. `./: pathmethodXxxReqXxxRes`

`// myapp/api/user.go`



```

        type UserUpdateReq struct {
            g.Meta      `summary:"" tags:"" `
            LoginName    string `json:"loginName" p:"loginName" v:"
passport" dc:"" `
            DisplayName string `json:"displayName" p:"displayName" dc:"" `
            Enabled      string `json:"enabled" p:"enabled" v:"in:enabled,
disabled" d:"enabled" dc:"" `
            Email        string `json:"email" p:"email" d:"" v:"email"
dc:"" `
            Phone        string `json:"phone" p:"phone" d:"" v:"phone"
dc:"" `
            Desc         string `json:"desc" p:"desc" d:"" v:"max-length:
255" dc:"" `
        }

        type UserGetRes struct {
            Uuid          string      `json:"uuid"          dc:"uuid" `
            LoginName     string      `json:"loginName"    dc:"" `
            DisplayName   string      `json:"displayName"  dc:"" `
            Email         string      `json:"email"        dc:"" `
            Phone         string      `json:"phone"        dc:"" `
            Enabled       string      `json:"enabled"      dc:"enabled"
disabled" `
            Desc         string      `json:"desc"         dc:"" `
            CreatedAt     *gtime.Time `json:"createdAt"    dc:"" `
            UpdatedAt     *gtime.Time `json:"updatedAt"    dc:"" `
        }

```

- UserUpdateReqgvalidate40emailphonepassport

pp -/\_/+ API

- UserUpdateReqjsonswaggerUI demoresponse
- g.Metaurl path;url method,swagger

4.

```
func Handler(ctx context.Context, req *Request) (res *Response, err
error)
```

```
req *Requestgf
```

5.

```
XxxRes
```

6.

```
APImiddleware
```

```

// myapp/internal/service/middleware.go
func (s *sMiddleware) ResponseHandler(r *ghttp.Request) {
    r.Middleware.Next()

    //
    if r.Response.BufferLength() > 0 {
        return
    }

    var (
        err error
        res interface{}
        code gcode.Code = gcode.CodeOK
    )
    res, err = r.GetHandlerResponse()
    if err != nil {

        code = gerror.Code(err)
        if code == errorCode.CodeNil {
            code = errorCode.CodeInternalServerError
        }
        if detail, ok := code.Detail().(errorCode.MyCodeDetail); ok {
            r.Response.WriteHeader(detail.HttpCode)
            r.Response.ClearBuffer() // gf responsehttp.StatusText
        }
        g.Log().Errorf(r.GetCtx(), "%+v", err)
        response.JsonExit(r, code.Code(), err.Error())
    } else {
        response.JsonExit(r, code.Code(), "", res)
    }
}

// myapp/utility/response/response.go

// Json JSON
func Json(r *ghttp.Request, code int, message string, data ...
interface{}) {
    var responseData interface{}
    if len(data) > 0 {
        responseData = data[0]
        if reflect.ValueOf(responseData).IsNil() { // responseData
interface
            responseData = g.Map{} // response nil{}
        }
    } else {
        responseData = g.Map{}
    }
    r.Response.WriteJson(JsonRes{
        Code:    code,
        Message: message,
        Data:    responseData,
    })
    r.Response.Header().Set("Content-Type", "application/json;
charset=utf-8") // response headcharset=utf-8
}

// JsonExit JSONHTTP
func JsonExit(r *ghttp.Request, code int, message string, data ...
interface{}) {
    Json(r, code, message, data...)
    r.Exit()
}

```

```

    JsonRes{
    Code:    code,
    Message: message,
    Data:    responseData,
    }

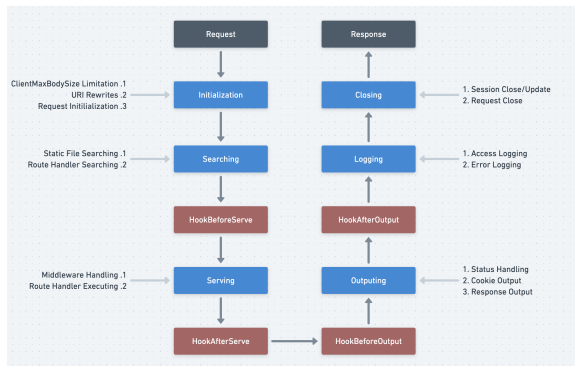
```

Content-TypeResponse.WriteHeader

Response.WriteHeaderfresponsehttp.StatusText r .Response .ClearBuffer ( )

### 3. middleware

gfmiddlewareResponseHandler middleware,hook



MiddlewareHOOKGF

1. `Server`
2. `Server`
3. `Server`

### 4. openAPIDoc(swagger)

Server/api.json/swagger/\*OpenAPIv3SwaggerUIopenapiPathswaggerPath

- `> v2.0.0-rc3swaggerUIredocredoctry it`
- `<= v2.0.0-rc3swaggerUIswaggerUI`

gfsswaggerUIgfsswaggerUI

- 1.

```

[server]
serverRoot    = "/resource/public" # swaggerUI
openapiPath   = "/api.json"
swaggerPath   = "/redoc" # /swagger/swagger

```

2. [swaggerUIgithubCDN](#)

-

```
    public
swagger
    favicon-16x16.png
    favicon-32x32.png
    index.css
    index.html
    oauth2-redirect.html
    swagger-initializer.js
    swagger-ui-bundle.js
    swagger-ui-bundle.js.map
    swagger-ui-es-bundle-core.js
    swagger-ui-es-bundle-core.js.map
    swagger-ui-es-bundle.js
    swagger-ui-es-bundle.js.map
    swagger-ui-standalone-preset.js
    swagger-ui-standalone-preset.js.map
    swagger-ui.css
    swagger-ui.css.map
    swagger-ui.js
    swagger-ui.js.map
```

swagger-initializer.js  
api.json

- CDNindex.html

```

        <!-- HTML for static distribution bundle build -->
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Swagger UI</title>
    <link rel="stylesheet" type="text/css" href="https://cdn.bootcdn.
net/ajax/libs/swagger-ui/4.10.3/swagger-ui.css" />
    <link rel="icon" type="image/png" href="https://cdn.jsdelivrivr.net
/npm/swagger-ui@4.10.3/dist/favicon-32x32.png" sizes="32x32" />
    <link rel="icon" type="image/png" href="https://cdn.jsdelivrivr.net
/npm/swagger-ui@4.10.3/dist/favicon-16x16.png" sizes="16x16" />
  </head>

  <body>
    <div id="swagger-ui"></div>
    <script src="https://cdn.bootcdn.net/ajax/libs/swagger-ui/4.10.3
/swagger-ui-bundle.js"></script>
    <script src="https://cdn.bootcdn.net/ajax/libs/swagger-ui/4.10.3
/swagger-ui-standalone-preset.js"></script>
    <script>
      window.onload = function() {
        //<editor-fold desc="Changeable Configuration Block">

        // the following lines will be replaced by docker
        /configurator, when it runs in a docker-container
        window.ui = SwaggerUIBundle({
          // url: "https://petstore.swagger.io/v2/swagger.json",
          url: "/api.json",
          dom_id: '#swagger-ui',
          deepLinking: true,
          presets: [
            SwaggerUIBundle.presets.apis,
            SwaggerUIStandalonePreset
          ],
          plugins: [
            SwaggerUIBundle.plugins.DownloadUrl
          ],
          layout: "StandaloneLayout"
        });

        //</editor-fold>
      };
    </script>

  </body>
</html>

```

3. <http://127.0.0.1:8199/swagger><http://127.0.0.1:8199/redoc>

## 5. controller

servicecontroller:

- controllerReq/Res service
- serviceInput/Output

## 6. service

serviceInput/Output

servicedaoorm

1. OmitEmpty()

/Insert, Replace, Update, Save map/struct nil,"",0 gdbmapstruct

```

        // UPDATE `user` SET `name`='john',update_time=null WHERE
`id`=1
db.Table("user").Data(g.Map{
    "name"      : "john",
    "update_time" : nil,
}).Where("id", 1).Update()

```

OmitEmpty

```

        // UPDATE `user` SET `name`='john' WHERE `id`=1
db.Table("user").OmitEmpty().Data(g.Map{
    "name"      : "john",
    "update_time" : nil,
}).Where("id", 1).Update()

```

omitemptyOmitEmpty

- structomitemptyjsonORMstruct
- omitemptyOmitEmptyORMstructomitemptyOmitEmptystructOmitEmptystruct

## 7. inputModeloutputModel

inputModeloutputModelcontrollerserviceInput/Outputmodel

## 8. dao/do/entity

gf gen daodao/do/entitygfv2.sql

```

[gfcli]
# DAO(database)
[[gfcli.gen.dao]]
    link    = "mysql:root:mysql@tcp(192.168.212.117:3306)/myapp" #
mysql
    descriptionTag = true # desription
    noModelComment = true #

```

11

gf gen daogfmysqldao/do/entity3dao

/internal/model/entity		
/internal/service/internal/do		
/internal/service/internal/dao		ORMentitydo

model

CLI model/entity CLICLI

servicemodel

daoDAODA0daoctxCtxTransaction

daodao/doserviceserviceadaodao

gfalmbic

## 9. errCode

errCode18Ncodei18n

codeHTTP200

- <https://stackoverflow.com/questions/27921537/returning-http-200-ok-with-error-within-response-body>

### 1. APIRESTAPI20errorcodemessage

- 200 200
- 201 POST 201
- 202
- 204 204
- 400 400
- 401 Token
- 403 Token
- 404
- 405 PATCH PUT PUT 405 PUT
- 409 409 body
- 500
- 502 A A B B A 502 B

12

### 2. REST API

- 200 200response

```
JsonRes{  
    Code:    code,  
    Message: message,  
    Data:    responseData,  
}
```

codegf

- 400
- 401
- 500 dbsql

412

### 3. APIRPC over HTTP200

RESTfulresponse

Code:

```

        type MyCode struct {
            code      int
            message string // message il8nkey
            detail    MyCodeDetail
        }
        type MyCodeDetail struct {
            HttpStatusCode int
        }

        func (c MyCode) MyDetail() MyCodeDetail {
            return c.detail
        }

        func (c MyCode) Code() int {
            return c.code
        }

        func (c MyCode) Message() string {
            return c.message
        }

        func (c MyCode) Detail() interface{} {
            return c.detail
        }

        func New(httpCode int, code int, message string) gcode.Code {
            return MyCode{
                code:      code,
                message: message,
                detail: MyCodeDetail{
                    HttpStatusCode: httpCode,
                },
            }
        }
    }

```

code

```

        var (
            // gfgithub.com\gogf\gf\v2@v2.0.0-rc2\errors\gcode\gcode.go
            CodeNil           = New(200, -1, "")
            CodeNotFound      = New(404, 65, "Not Found")
            CodeInternalError = New(500, 50, "An error occurred internally")

            // 10000
            CodeBadRequest = New(400, 10000, `{#badRequestParameter}`)

            //20000
            UserNotFound      = New(404, 20001, `{#userNotExists}`)
            LoginNameConflicted = New(403, 20002, `{#loginNameConflicted}`)

            // 30000
            DesktopNotFound = New(404, 30001, `{#desktopNotExists}`)
        )
    }

```

codeil8n



```

        func NewMyErr(ctx context.Context, code gcode.Code, params ...
string) error {
    // g.I18n().Tf(ctx context.Context, format string, values ...interface
{}) values[]interface{},
    v := make([]interface{}, len(params))
    for i, p := range params {
        v[i] = p
    }
    tfStr := g.I18n().Tf(ctx, code.Message(), v...)
    return gerror.NewCode(code, tfStr)
}

```

```
errorCode.NewMyErr(ctx, errorCode.DesktopNotFound, req.Uuid)i18n
```

```
// zh-CN.toml
desktopNotExists = " uuid=%s "
```

```
// en.toml
desktopNotExists = "Desktop uuid=%s not exists"
```

## 10. I18N

i18n gfi18n

i18ntoml

i18nmain.goi18n/mapmapvalueg.I18n().SetPath("resource/i18n")i18ngres

TODO: i18nresource/i18n

## 11.

yaml/toml/json/xml/initoml.

g.Cfg()

manifest/config/config.toml

## server

[gf server config](#)

```

[server]
#
address          = ":80"                # ":80"
httpsAddr        = ":443"                # TLS/HTTPS
httpsCertPath    = ""                    # TLS/HTTPS
httpsKeyPath     = ""                    # TLS/HTTPS
readTimeout     = "60s"                  # 60
writeTimeout    = "0"                    # 0
idleTimeout     = "60s"                  # Keep-Alive60
maxHeaderBytes  = "10240"                 # HeaderByte10KB
keepAlive       = true                    # Keep-Alivetrue
serverAgent     = "GoFrame HTTP Server"  # Agent"GoFrame
HTTP Server"

#
indexFiles      = ["index.html","index.htm"] # ["index.html",
"index.htm"]
indexFolder     = false                    # 403
serverRoot     = "/var/www"                #
searchPaths    = ["/home/www","/var/lib/www"] #
fileServerEnabled = false                  # false

```

```

# Cookie
cookieMaxAge      = "365d"          # Cookie365
cookiePath        = "/"            # Cookie"/"
cookieDomain      = ""             # CookieCookie

# Sessions
sessionMaxAge     = "24h"          # Session24
sessionIdName     = "gfsessionId"  # SessionIdgfsessionId
sessionCookieOutput = true         # SessionSessionIdCookie>true
sessionPath       = "/tmp/gsessions" # Sessiongsessions

# Logging
logPath           = ""             #
logStdout         = true           # true
errorStack        = true           # Server>true
errorLogEnabled   = true           # true
errorLogPattern   = "error-{Ymd}.log" # "error-{Ymd}.log"
accessLogEnabled  = false          # false
accessLogPattern  = "access-{Ymd}.log" # "access-{Ymd}.log"

# ()
[server.logger]
  path=           "/var/log/"      #
  file=           "{Y-m-d}.log"    # "{Y-m-d}.log"
  prefix=         ""              #
  level=          "all"            #
  stdout=         true            # true
  rotateSize=     0                # 0
  rotateExpire=   0                # 0
  rotateBackupLimit= 0            # 0
  rotateBackupExpire= 0            # 0
  rotateBackupCompress= 0          # 0-90
  rotateCheckInterval= "1h"        # 1

# PProf
pprofEnabled      = false          # PProffalse
pprofPattern      = ""            # PProfPProfServer

#
clientMaxBodySize = 810241024      # Body(Byte)8*1024*1024=8MB
formParsingMemory = 1048576        # (Byte)1024*1024=1MB
nameToUriType     = 0              # 00/1/2/3
                                # 0'- '
                                # 1: URI.
                                # 2: .
                                # 3: .
                                # false
routeOverWrite    = false          # false
dumpRouterMap     = true           # Server>true
graceful          = false          # 10000TCP>false
gracefulTimeout   = 2              # 2<br />

```

db

```

[database]:
:
host="127.0.0.1" #
port="3306"      #
user="root"      #
pass="mysql"     #
name="mysql"     #
type="mysql"     # (mysql/pgsql/mssql/sqlite/oracle)
link = "mysql:root:mysql@tcp(192.168.212.117:3306)/myapp" # ()(Host,
Port,User,Pass,Name)type;typelink
role="master"    # ()(master/slave)master
debug=true       # ()
prefix="gf_"     # ()
dryRun=true      # ()ORM()
charset="utf8"   # ()(: utf8/gbk/gb2312)utf8
weight=100       # ()
timezone="local" # ():local
maxIdle=10       # ()
maxOpen=100      # ()
maxLifetime="30s" # ()
createdAt="createdAt" # ()
updatedAt="updatedAt" # ()
deletedAt="deletedAt" # ()
timeMaintainDisabled=true # ()trueCreatedAt/UpdatedAt/DeletedAt

```

- :gdb
- gdblog.LoggergdbDEBUGDEBUGdebugtrue.

```

[database]
[database.logger]
path = "/var/log/gf-app/sql"
level = "all"
stdout = true
[database.default]
link = "mysql:root:12345678@tcp(127.0.0.1:3306)/user_center"
debug = true

```

database.loggergdb.

## log

```

[logger]
path= "/var/log/" #
file= "{Y-m-d}.log" # "{Y-m-d}.log"
prefix= "" #
level= "all" #
ctxKeys= [] # ContextContext
header= true # true
stdout= true # true
rotateSize= 0 # 0
rotateExpire= 0 # 0
rotateBackupLimit= 0 # 0
rotateBackupExpire= 0 # 0
rotateBackupCompress= 0 # 0-90
rotateCheckInterval= "1h" # 1
stdoutColorDisabled= false #
writerColorEnable= false # false

```

levelDEBU < INFO < NOTI < WARN < ERRO < CRITALL, DEV, PRODlevel

log

```
    logger:
  path:    "/var/log"
  level:   "all"
  stdout:  false
  logger1:
    path:   "/var/log/logger1"
    level:  "dev"
    stdout: false
  logger2:
    path:   "/var/log/logger2"
    level:  "prod"
    stdout: true
```

Logger

```
    // logger.logger1
11 := g.Log("logger1")
// logger.logger2
12 := g.Log("logger2")
// logger
13 := g.Log("none")
// logger
14 := g.Log()
```

log gfserverdbloglogloggfsver

```
2022-05-07 11:35:20.949 [DEBU] SetServerRoot path: /root/go-
workspace/MyGoFrame/myapp/resource/public
2022-05-07 11:35:20.958 [INFO] swagger ui is serving at address: http://0.
0.0.0:8199/redoc/
2022-05-07 11:35:20.958 [INFO] openapi specification is serving at
address: http://0.0.0.0:8199/api.json
2022-05-07 11:35:20.959 [INFO] pid[22663]: http server started listening
on [0.0.0.0:8199]
```

ADDRESS	METHOD	ROUTE	MIDDLEWARE
HANDLER			
-----	-----	-----	-----
0.0.0.0:8199	ALL	/*	github.com/gogf/gf/v2/net
/ghttp.internalMiddlewareServerTracing			GLOBAL MIDDLEWARE
-----	-----	-----	-----
.....			

4glog

4loggfconfluencegf

1. gfhandlerhandlerlgf

v2.1masterglog.SetDefaultHandler

2. loglogger

```
gf-build
[gfcli.build]
  name=      "my-goframe"
  arch=      "all"
  system=    "all"
  mod=       "none"
  cgo=       0
  pack=      "manifest/config/resource/i18n"
  version=   ""
  output=    "./bin"
  extra=     ""
```

name		go	windows.exe	gf
arch			,all	386,amd64,arm
system			,all	linux,darwin,windows
path		./bin		./bin
mod			go build -mod	none
cgo		0	CGO	0
pack			,internal/packed/data.go	public,template
version				v1.0.0
output			namepath	./bin/gf.exe
extra			go build	
varMap				gfcli: build: name: "gf" arch: "all" system: "all" mod: "none" cgo: 0 pack: "" version: "v1.0.0" output: "./bin" extra: "" varMap: k1: v1 k2: v2

```
gf-cli
[[gfcli.gen.dao]]
  link      = "mysql:root:mysql@tcp(192.168.212.117:3306)/myapp"
  descriptionTag = true
  noModelComment = true
  removePrefix = "gf_"
```

gfcli.gen.dao			dao	-
link			mysql, postgresql, dsn <a href="#">ORM</a>	-
path		internal	daomodel	./app
group		default		default``order``user
prefix				order_``user_

removePrefix			,	gf_
tables				user, user_detail
tablesEx			Tables Excluding	product, order
jsonCase		CamelLower	modeljsonCamelCamelLowerSnakeSnakeScreamingSnakeFirstUpperKebabKebabScreaming	Snake
stdTime		false	time.Time*gtime.Time	true
gJsonSupport		false	JSON*gjson.Json	true
overwriteDao		false	daodao/internaldao/internal	true
importPrefix		go.mod	Goimportgen dao	<a href="https://github.com/gogf/gf">github.com/gogf/gf</a>
descriptionTag		false	desription	true
noModelComment		false		true

## 12.

gfgo web testify+gclient

```
package test

import (
    "context"
    "encoding/json"
    "fmt"
    "testing"

    "github.com/gogf/gf/v2/frame/g"

    "github.com/stretchr/testify/suite"
)

type Result struct {
    Code    int
    Message string
    Data    interface{}
}

type User struct {
    Uuid       string
    UserName   string
    DisplayName string
    Email      string
    Phone      string
    Desc       string
}

var (
    ctx    = context.TODO()
    client = g.Client()
    admin  = g.Map{
        "userName": "admin",
        "password": "password",
    }
    userData = g.Map{
        "userName": "zhangsanzhangsan",
        "displayName": "",
        "email": "san.zhang@gmail.com",
        "phone": "17628272827",
        "password": "123qwe.",
        "desc": "zhang",
    }
    userStruct = &User{}
    resultStruct = &Result{}
)
```

```

    uerUuid      = ""
    token        = ""
)

type MyTestSuit struct {
    suite.Suite
}

func (s *MyTestSuit) SetupSuite() {
    client.SetPrefix(fmt.Sprintf("http://127.0.0.1:%s", "8199"))
    s.login()
    client.SetHeader("Authorization", "Bearer "+token)
    fmt.Println("SetupSuiteconfig http client and get token before all
test")
}

func (s *MyTestSuit) login() {
    getContentStr := client.PostContent(ctx, "/login", admin)
    json.Unmarshal([]byte(getContentStr), resultStruct)
    s.Assert().Equal(resultStruct.Code, 0)
    s.Assert().Equal(resultStruct.Data.(map[string]interface{})["user"].
(map[string]interface{})["userName"], admin["userName"])
    token = fmt.Sprint(resultStruct.Data.(map[string]interface{})["token"])
}

func (s *MyTestSuit) logout() {
    getContentStr := client.DeleteContent(ctx, "/logout/"+uerUuid)
    json.Unmarshal([]byte(getContentStr), resultStruct)
    s.Assert().Equal(resultStruct.Code, 0)
}

func (s *MyTestSuit) TearDownSuite() {
    fmt.Println("TearDownSuite delete token after all test")
}

func (s *MyTestSuit) SetupTest() {
}

func (s *MyTestSuit) TearDownTest() {
}

func (s *MyTestSuit) BeforeTest(suiteName, testName string) {
}

func (s *MyTestSuit) AfterTest(suiteName, testName string) {
}

func (s *MyTestSuit) TestUserCRUD() {
    //create user
    createContentStr := client.PostContent(ctx, "/users", userData)
    json.Unmarshal([]byte(createContentStr), resultStruct)
    s.Assert().Equal(resultStruct.Code, 0)
    s.Assert().Equal(resultStruct.Data.(map[string]interface{})
["userName"], userData["userName"])
    uerUuid = resultStruct.Data.(map[string]interface{})["uuid"].(string)
    // GET user
    getContentStr := client.GetContent(ctx, "/users/"+uerUuid)
    json.Unmarshal([]byte(getContentStr), resultStruct)
    s.Assert().Equal(resultStruct.Code, 0)
    s.Assert().Equal(resultStruct.Data.(map[string]interface{})
["userName"], userData["userName"])
    // list user
    listContentStr := client.GetContent(ctx, "/users")
    json.Unmarshal([]byte(listContentStr), resultStruct)
    s.Assert().Equal(resultStruct.Code, 0)
    s.Assert().Greater(resultStruct.Data.(map[string]interface{})
["total"], float64(0))
    // update user
    updateContentStr := client.PatchContent(ctx, "/users/"+uerUuid, g.Map
{"displayName": "wangmazi"})
    json.Unmarshal([]byte(updateContentStr), resultStruct)

```

```

    s.Assert().Equal(resultStruct.Code, 0)
    s.Assert().Equal(resultStruct.Data.(map[string]interface{}))
["displayName", "wangmazi")
    // delete user
    deleteContentStr := client.DeleteContent(ctx, "/users/"+uerUuid)
    json.Unmarshal([]byte(deleteContentStr), resultStruct)
    s.Assert().Equal(resultStruct.Code, 0)
}

func TestExample(t *testing.T) {
    suite.Run(t, new(MyTestSuit))
}

```

```

=== RUN    TestExample
SetupSuiteconfig http client and get token before all test
--- PASS: TestExample (0.79s)
=== RUN    TestExample/TestUserCRUD
TearDownSuitedelete token after all test
--- PASS: TestExample/TestUserCRUD (0.62s)
PASS

```

## 13. token

- [httpheaderAuthorization](#)
  - [Basic](#)
  - [Bearer](#)
  - [RFC 7235](#) and [HTTP Authentication Scheme Registry](#)
- [API keys](#) [requestheadercookiesurl](#)
- [OAuth 2](#)
- [OpenID Connect Discovery](#)

JWThttp Bearer

[gtoken](#),

1. redis
2. jwt
3. jwt
4. jwt
5. jwt

## 14.

systemctlSystemd -

1. shell[GoFrame](#)
2. systemctlmygf-app.service



```

[Unit]
Description=my-goframe app daemon
After=mysql.service
Wants=redis-server.service mysql.service

[Service]
Type=forking
ExecStart=/opt/my-goframe/run.sh start
ExecStop=/opt/my-goframe/run.sh stop
#Restart=always
#RestartSec=5s
#StartLimitInterval=0

[Install]
WantedBy=multi-user.target

```

mysqlmysqlrediswants redis

```

#Restart=always
#RestartSec=5s
#StartLimitInterval=0

```

3. mygf-app.servicewen\lib\systemd\system\
4. systemctl enable mygf-app.service
5. systemctl

deb

deb

1. deb postinst(postinstallation) postrm(postremove)preinst(preinstallation)prerm(preremove)
  - a. postinst

```

#!/bin/bash

#####
# Deb
#####

#
systemctl daemon-reload
systemctl enable mygf-app.service

echo "postinst done"
exit 0

```

- b. postrm

```
#!/bin/sh

#####
#
#####

echo "postrm done"
# dpkg
rm -rf /opt/my-goframe
exit 0
```

- c.
- 2. dpkg-deb -b xxxdpkg-debdeb
- 3. shell

debdpkg -isystemctlperfect ~

1. Dockerfile

```
FROM loads/alpine:3.8
LABEL maintainer="xiaobin.zhao@cstack.io"

#####
#####
#
# INSTALLATION
#####
#####

#
ENV WORKDIR /opt/my-goFrame

#
COPY ./bin/linux_amd64/mygf-app $WORKDIR/mygf-app
RUN chmod +x $WORKDIR/mygf-app

# TODO: gf buildpack,
COPY ./resource/il8n $WORKDIR/resource/il8n
COPY ./resource/public $WORKDIR/resource/public

#####
#####
#
# START
#####
#####
WORKDIR $WORKDIR
CMD ./mygf-app
```

- 2. gf buildpack
- 3. gf dockergf bulddocker build
- 4. docker run

15.

jmeterSQL

					QPS	error %503 Service Unavailable
100	10000	158	22	19214	400.5	0.02%
50	10000	85	27	7080	461.8	0
20	10000	30	27	259	616.8	0
10	10000	31	27	505	301.4	0

qps

fastapi

limit_concurrency=10						
					QPS	error %503 Service Unavailable
5	1000	38	31	113	116.2925922	0
10	1000	39	28	86	205.3809817	0.121
15	1050	43	28	81	242.2145329	0.29047619
20	1000	46	28	81	308.9280198	0.412
50	1000	52	28	93	485.9086492	0.637
100	1000	95	28	1082	425.3509145	0.641
200	1000	56	28	88	756.429652	0.785
1000	1000	104	54	317	823.723229	0.878

gopython

...