-gen dao

gen daoCLIdaodoentityGo

gf gen dao

```
$ gf gen dao -h
USAGE
    gf gen dao [OPTION]

OPTION
    -p, --path                  directory path for generated files
    -l, --link                  database configuration, the same as the
ORM configuration of GoFrame
    -t, --tables                generate models only for given tables,
multiple table names separated with ','
    -x, --tablesEx              generate models excluding given tables,
multiple table names separated with ','
    -g, --group                 specifying the configuration group name of
database for generated ORM instance,
                                it's not necessary and the default value
is "default"
    -f, --prefix                add prefix for all table of specified link
/database tables
    -r, --removePrefix          remove specified prefix of the table,
multiple prefix separated with ','
    -rf, --removeFieldPrefix    remove specified prefix of the field,
multiple prefix separated with ','
    -j, --jsonCase              generated json tag case for model struct,
cases are as follows:
                                | Case            | Example            |
                                |---------------- |--------------------|
                                | Camel           | AnyKindOfString    |
                                | CamelLower      | anyKindOfString    |
default
                                | Snake           | any_kind_of_string |
                                | SnakeScreaming  | ANY_KIND_OF_STRING |
                                | SnakeFirstUpper | rgb_code_md5       |
                                | Kebab           | any-kind-of-string |
                                | KebabScreaming  | ANY-KIND-OF-STRING |
    -i, --importPrefix          custom import prefix for generated go files
    -d, --daoPath               directory path for storing generated dao
files under path
    -o, --doPath                directory path for storing generated do
files under path
    -e, --entityPath            directory path for storing generated
entity files under path
    -t1, --tplDaoIndexPath      template file path for dao index file
    -t2, --tplDaoInternalPath   template file path for dao internal file
    -t3, --tplDaoDoPath         template file path for dao do file
    -t4, --tplDaoEntityPath     template file path for dao entity file
    -s, --stdTime               use time.Time from stdlib instead of gtime.
Time for generated time/date fields of tables
    -w, --withTime              add created time for auto produced go files
    -n, --gJsonSupport          use gJsonSupport to use *gjson.Json
instead of string for generated json fields of
                                tables
    -v, --overwriteDao          overwrite all dao files both inside
/outside internal folder
    -c, --descriptionTag        add comment to description tag for each
field
    -k, --noJsonTag             no json tag will be added for each field
```

Content Menu

```
    -m, --noModelComment        no model comment will be added for each
field
    -a, --clear                 delete all generated go files that do not
exist in database
    -y, --typeMapping           custom local type mapping for generated
struct attributes relevant to fields of table
    -h, --help                  more information about this command

EXAMPLE
    gf gen dao
    gf gen dao -l "mysql:root:12345678@tcp(127.0.0.1:3306)/test"
    gf gen dao -p ./model -g user-center -t user,user_detail,user_login
    gf gen dao -r user_

CONFIGURATION SUPPORT
    Options are also supported by configuration file.
    It's suggested using configuration file instead of command line
arguments making producing.
    The configuration node name is "gfcli.gen.dao", which also supports
multiple databases, for example(config.yaml):
    gfcli:
      gen:
        dao:
        - link:     "mysql:root:12345678@tcp(127.0.0.1:3306)/test"
          tables:   "order,products"
          jsonCase: "CamelLower"
        - link:   "mysql:root:12345678@tcp(127.0.0.1:3306)/primary"
          path:   "./my-app"
          prefix: "primary_"
          tables: "user, userDetail"
          typeMapping:
            decimal:
              type:   decimal.Decimal
              import: github.com/shopspring/decimal
            numeric:
              type: string
```

```
makemake dao
```

/hack/config.yaml

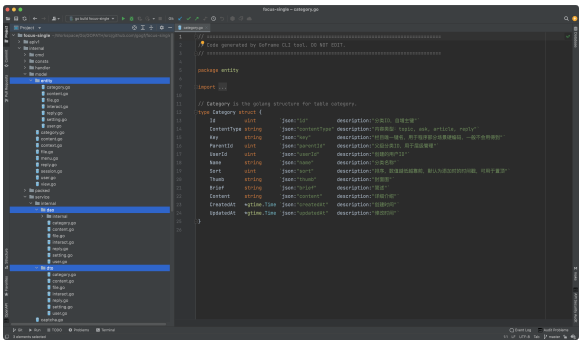
```
gfcli:
  gen:
    dao:
    - link:     "mysql:root:12345678@tcp(127.0.0.1:3306)/test"
      tables:   "order,products"
      jsonCase: "CamelLower"

    - link:   "mysql:root:12345678@tcp(127.0.0.1:3306)/primary"
      path:   "./my-app"
      prefix: "primary_"
      tables: "user, userDetail"

        # sqlitesqlitegfgf\cmd\gf\internal\cmd\cmd_gen_dao.goimportsqlite
gcc
        - link: "sqlite:./file.db"
```


gfcli.gen.dao		dao	-
link		mysql, postgresql, dsn ORM	-
path	internal	daomodel	./app
group	default		default order user
prefix			order_ user_
removePrefix		,	gf_
removeFieldPrefix		,	f_
tables			user, user_detail
tablesEx		Tables Excluding	product, order
jsonCase	CamelLower	modeljsonCamelCamelLowerSnakeSnakeScreamingSnakeFirstUpperKebabKebabScreaming	Snake
stdTime	false	time.Time*gtime.Time	true
withTime	false		
gJsonSupport	false	JSON*gjson.Json	true
overwriteDao	false	daodao/internaldao/internal	true
importPrefix	go.mod	Goimportgen dao	github.com/gogf/gf
descriptionTag	false	desription	true
noJsonTag	false	json	
noModelComment	false		true
clear	false	dao/do/entity	

typeMapping	decimal:
	  type: float64
	money:
	  type: float64
	numeric:
	  type: float64
	smallmoney:
	  type: float64
		v2.5
		Goimport
		decimal:
		  type: decimal.Decimal
		  import: github.com/shopspring/decimal	

daoPath	dao	DAO	
doPath	model/do	DO	
entityPath	model/entity	Entity	
tplDaoIndexPath		DAO Index	
tplDaoInternalPath		DAO Internal	
tplDaoDoPath		DO	
tplDaoEntityPath		Entity	

https://github.com/gogf/focus-single



13dao

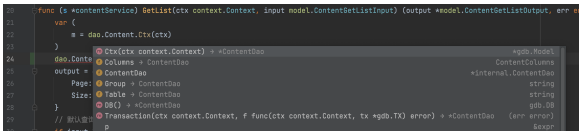
/internal/dao		ORMentitydo
/internal/model/do		do • • DAO- • do
/internal/model/entity		

2model

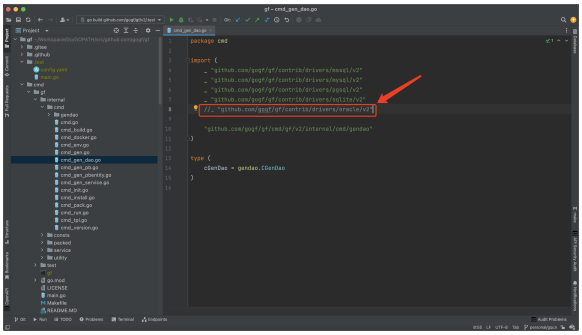
CLI model/entity CLICLI

servicemodel

3daoDAODAOdaoctxCtxTransaction



gen daoOracleCLICGO



bool

boolbit(1)boolgen daobit(1)booltinyint(1)bool