

# gen dao

daodaomodel

gf gen dao

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

OPTION
--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 ','
-g, --group     specifying the configuration group name of database for generated ORM instance,
               it's not necessary and the default value is "default"
-p, --prefix    add prefix for all table of specified link/database tables.
-r, --removePrefix remove specified prefix of the table, multiple prefix separated with ','
-m, --mod       module name for generated golang file imports.
-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 |
--tplDaoIndex   template content for Dao index files generating.
--tplDaoInternal template content for Dao internal files generating.
--tplModelIndex template content for Model index files generating.
--tplModelInternal template content for Model internal files generating.
```

## 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 "gf.gen.dao", which also supports multiple databases, for example:

```
[gfcli]
[[gfcli.gen.dao]]
  link    = "mysql:root:12345678@tcp(127.0.0.1:3306)/test"
  tables  = "order,products"
  jsonCase = "CamelLower"
[[gfcli.gen.dao]]
  link    = "mysql:root:12345678@tcp(127.0.0.1:3306)/primary"
  path    = "./my-app"
  prefix  = "primary_"
  tables  = "user, userDetail"
```

## EXAMPLES

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

```
[gfcli]
[[gfcli.gen.dao]]
  link    = "mysql:root:12345678@tcp(127.0.0.1:3306)/order"
  group   = "order"
  prefix  = "order_"
  tables  = ""
[[gfcli.gen.dao]]
  link    = "mysql:root:12345678@tcp(127.0.0.1:3306)/user"
  group   = "user"
  prefix  = "user_"
  tables  = "user,userDetail,userScore"
```

gfcli.gen.dao			dao	-
link			mysql, postgresql, dsn <a href="#">ORM</a>	mysql:root:12345678@tcp(127.0.0.1:3306)/user
mod			goimportgo.mod	<a href="#">github.com/gogf/gf-demos</a>
group		default		default order user
tables				user, userDetail
path		./app	daomodel	./app
prefix				order_ user_
removePrefix				gf_
jsonCase		CamelLower	model jsonCamelCamelLowerSnakeSnakeScreamingSnakeFirstUpperKebabKebabScreaming	CamelLower

1. dao/internal model/internal internaldaomodel
2. dao service
3. model