

GoFrame



`g.Log()` `Logger` `Logger` `logger` . `logger` <https://pkg.go.dev/github.com/gogf/gf/v2/os/glog#Config>

Content Menu

- - [1](#)
 - [2](#)
-
-

```
logger:
  path:      "/var/log/"      #
  file:      "{Y-m-d}.log"    # "{Y-m-d}.log"
  prefix:    ""               #
  level:     "all"            #
  timeFormat: "2006-01-02T15:04:05" # Golang
  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
```

`level` `DEBU` < `INFO` < `NOTI` < `WARN` < `ERRO` < `CRITALL`, `DEV`, `PROD` `level` -

1

```
logger:
  path:      "/var/log"
  level:     "all"
  stdout:    false
```

`g.Log()`

2

`Logger`

```
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
l1 := g.Log("logger1")
// logger.logger2
l2 := g.Log("logger2")
// logger
l3 := g.Log("none")
// logger
l4 := g.Log()
```

glog

1. SetConfigSetConfigWithMap
2. LoggerSet*
3. Logger

SetConfigWithMapKey-Value/LoggerKeyConfigstruct-/_/ -Struct

```
logger := glog.New()
logger.SetConfigWithMap(g.Map{
    "path":    "/var/log",
    "level":   "all",
    "stdout":  false,
    "StStatus": 0,
})
logger.Print("test")
```

StStatus0stStatus, st-status, st_status, St Status

HTTP ServerGRPC ServerORM

GoFrameg.Log()glog.New()API