

GF

```
g.Log()g.Log()
```

```
g.Log()LoggerLoggerlogger.loggerhttps://godoc.org/github.com/gogf/gf/os/glog#Config
```

```
[logger]
    Path      = "/var/log/"      #
    File       = "{Y-m-d}.log"   # "{Y-m-d}.log"
    Prefix     = ""              #
    CtxKeys    = []              # ContextContext
    HeaderPrint = true           # true
    StdoutPrint = true           # true
    RotateSize  = 0              # 0
    RotateExpire = 0             # 0
    RotateBackupLimit = 0        # 0
    RotateBackupExpire = 0       # 0
    RotateBackupCompress = 0     # 0-90
    RotateCheckInterval = "1h"   # 1
```

1

```
[logger]
    path  = "/var/log"
    level = "all"
    stdout = false
```

```
g.Log()
```

2

Logger

```
[logger]
    path  = "/var/log"
    level = "all"
    stdout = false
    [logger.logger1]
        path  = "/var/log/logger1"
        level = "dev"
        stdout = false
    [logger.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()
```

glog

<https://godoc.org/github.com/gogf/gf/os/glog>

Content Menu

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

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

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