

-Writer



WriterIOHandler-Handler

Content Menu

- [Writer](#)
- [1HOOK](#)
- [2graylog](#)

Writer

glogio.Writerio.Writer

```
type Writer interface {
    Write(p []byte) (n int, err error)
}
```

SetWriterToWriterWriter

glog.Loggerio.Writerglog

1HOOK

WriterMyWriterWriterPANIFATAHTTPMonitorglog

```
package main

import (
    "context"
    "fmt"

    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/net/ghttp"
    "github.com/gogf/gf/v2/os/glog"
    "github.com/gogf/gf/v2/text/gregex"
)

type MyWriter struct {
    logger *glog.Logger
}

func (w *MyWriter) Write(p []byte) (n int, err error) {
    var (
        s    = string(p)
        ctx = context.Background()
    )
    if gregex.IsMatchString(`PANI|FATA`, s) {
        fmt.Println("SERIOUS ISSUE OCCURRED!! I'd better tell
monitor in first time!")
        g.Client().PostContent(ctx, "http://monitor.mydomain.com",
s)
    }
    return w.logger.Write(p)
}

func main() {
    var ctx = context.Background()
    glog.SetWriter(&MyWriter{
        logger: glog.New(),
    })
    glog.Fatal(ctx, "FATAL ERROR")
}
```

```
SERIOUS ISSUE OCCURRED!! I'd better tell monitor in first time!
2019-05-23 20:14:49.374 [FATA] FATAL ERROR
Stack:
1. /Users/john/Workspace/Go/GOPATH/src/github.com/gogf/gf/v2/geg/os/glog
/glog_writer_hook.go:27
```

2graylog

GraylogWriter

Graylog ELK Dashboard

```
package main

import (
    "context"

    "github.com/gogf/gf/v2/os/glog"
    "github.com/robertkowalski/graylog-golang"
)

type MyGrayLogWriter struct {
    gelf    *gelf.Gelf
    logger  *glog.Logger
}

func (w *MyGrayLogWriter) Write(p []byte) (n int, err error) {
    w.gelf.Send(p)
    return w.logger.Write(p)
}

func main() {
    var ctx = context.Background()
    glog.SetWriter(&MyGrayLogWriter{
        logger : glog.New(),
        gelf    : gelf.New(gelf.Config{
            GraylogPort      : 80,
            GraylogHostname : "graylog-host.com",
            Connection       : "wan",
            MaxChunkSizeWan : 42,
            MaxChunkSizeLan : 1337,
        }),
    })
    glog.Println(ctx, "test log")
}
```