

gloggoroutine

SetAsync/SetFlagsAsync



Content Menu

- [SetAsync](#)
- [Async](#)

SetAsync

SetAsync

```
package main

import (
    "context"
    "time"

    "github.com/gogf/gf/v2/frame/g"
)

func main() {
    ctx := context.TODO()
    g.Log().SetAsync(true)
    for i := 0; i < 10; i++ {
        g.Log().Print(ctx, "async log", i)
    }
}
```

```
package main

import (
    "context"
    "time"

    "github.com/gogf/gf/v2/frame/g"
)

func main() {
    ctx := context.TODO()
    g.Log().SetAsync(true)
    for i := 0; i < 10; i++ {
        g.Log().Print(ctx, "async log", i)
    }
    time.Sleep(time.Second)
}
```

```
2019-06-02 15:44:21.399 async log 0
2019-06-02 15:44:21.399 async log 1
2019-06-02 15:44:21.399 async log 2
2019-06-02 15:44:21.399 async log 3
2019-06-02 15:44:21.399 async log 4
2019-06-02 15:44:21.399 async log 5
2019-06-02 15:44:21.399 async log 6
2019-06-02 15:44:21.399 async log 7
2019-06-02 15:44:21.399 async log 8
2019-06-02 15:44:21.399 async log 9
```

Async

```
package main

import (
    "context"
    "time"

    "github.com/gogf/gf/v2/frame/g"
)

func main() {
    ctx := context.TODO()
    for i := 0; i < 10; i++ {
        g.Log().Async().Print(ctx, "async log", i)
    }
    g.Log().Print(ctx, "normal log")
    g.Log().Print(ctx, "normal log")
    g.Log().Print(ctx, "normal log")
    time.Sleep(time.Second)
}
```

```
2022-01-05 15:00:44.101 normal log
2022-01-05 15:00:44.101 async log 0
2022-01-05 15:00:44.101 async log 1
2022-01-05 15:00:44.101 async log 2
2022-01-05 15:00:44.101 async log 3
2022-01-05 15:00:44.101 async log 4
2022-01-05 15:00:44.101 async log 5
2022-01-05 15:00:44.101 async log 6
2022-01-05 15:00:44.101 async log 7
2022-01-05 15:00:44.101 async log 8
2022-01-05 15:00:44.101 async log 9
2022-01-05 15:00:44.101 normal log
2022-01-05 15:00:44.103 normal log
```