-gmlock

Try*Lock

> *Remove*


```
import "github.com/gogf/gf/v2/os/gmlock"
```

https://pkg.go.dev/github.com/gogf/gf/v2/os/gmlock

```
func Lock(key string)
func LockFunc(key string, f func())
func RLock(key string)
func RLockFunc(key string, f func())
func RUnlock(key string)
func Remove(key string)
func TryLock(key string) bool
func TryLockFunc(key string, f func()) bool
func TryRLock(key string) bool
func TryRLockFunc(key string, f func()) bool
func Unlock(key string)
type Locker
    func New() *Locker
    func (l *Locker) Clear()
    func (l *Locker) Lock(key string)
    func (l *Locker) LockFunc(key string, f func())
    func (l *Locker) RLock(key string)
    func (l *Locker) RLockFunc(key string, f func())
    func (l *Locker) RUnlock(key string)
    func (l *Locker) Remove(key string)
    func (l *Locker) TryLock(key string) bool
    func (l *Locker) TryLockFunc(key string, f func()) bool
    func (l *Locker) TryRLock(key string) bool
    func (l *Locker) TryRLockFunc(key string, f func()) bool
    func (l *Locker) Unlock(key string)
```

1

```
package main

import (
    "time"
    "sync"
    "github.com/gogf/gf/v2/os/glog"
    "github.com/gogf/gf/v2/os/gmlock"
)

func main() {
    key := "lock"
    wg  := sync.WaitGroup{}
    for i := 0; i < 10; i++ {
        wg.Add(1)
        go func(i int) {
            gmlock.Lock(key)
            glog.Println(i)
            time.Sleep(time.Second)
            gmlock.Unlock(key)
            wg.Done()
        }(i)
    }
    wg.Wait()
}
```

10goroutinegoroutinegoroutine1goroutine

```
2018-10-15 23:57:28.295 9
2018-10-15 23:57:29.296 0
2018-10-15 23:57:30.296 1
2018-10-15 23:57:31.296 2
2018-10-15 23:57:32.296 3
2018-10-15 23:57:33.297 4
2018-10-15 23:57:34.297 5
2018-10-15 23:57:35.297 6
2018-10-15 23:57:36.298 7
2018-10-15 23:57:37.298 8
```

2TryLock

TryLocktruegoroutinefalse

```
package main

import (
    "sync"
    "github.com/gogf/gf/v2/os/glog"
    "time"
    "github.com/gogf/gf/v2/os/gmlock"
)

func main() {
    key := "lock"
    wg  := sync.WaitGroup{}
    for i := 0; i < 10; i++ {
        wg.Add(1)
        go func(i int) {
            if gmlock.TryLock(key) {
                glog.Println(i)
                time.Sleep(time.Second)
                gmlock.Unlock(key)
            } else {
                glog.Println(false)
            }
            wg.Done()
        }(i)
    }
    wg.Wait()
}
```

1goroutinegoroutineTryLock

```
2018-10-16 00:01:59.172 9
2018-10-16 00:01:59.172 false
2018-10-16 00:01:59.172 false
2018-10-16 00:01:59.172 false
2018-10-16 00:01:59.172 false
2018-10-16 00:01:59.172 false
2018-10-16 00:01:59.172 false
2018-10-16 00:01:59.172 false
2018-10-16 00:01:59.172 false
2018-10-16 00:01:59.176 false
```