

TCP-

gtcpgtcp.PoolConn600

```
import "github.com/gogf/gf/v2/net/gtcp"
```

Content Menu

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

<https://pkg.go.dev/github.com/gogf/gf/v2/net/gtcp>

```
type PoolConn
func NewPoolConn(addr string, timeout ...int) (*PoolConn, error)
func (c *PoolConn) Close() error
func (c *PoolConn) Recv(length int, retry ...Retry) ([]byte, error)
func (c *PoolConn) RecvLine(retry ...Retry) ([]byte, error)
func (c *PoolConn) RecvPkg(option ...PkgOption) ([]byte, error)
func (c *PoolConn) RecvPkgWithTimeout(timeout time.Duration, option ...
PkgOption) ([]byte, error)
func (c *PoolConn) RecvWithTimeout(length int, timeout time.Duration,
retry ...Retry) (data []byte, err error)
func (c *PoolConn) Send(data []byte, retry ...Retry) error
func (c *PoolConn) SendPkg(data []byte, option ...PkgOption) (err
error)
func (c *PoolConn) SendPkgWithTimeout(data []byte, timeout time.
Duration, option ...PkgOption) error
func (c *PoolConn) SendRecv(data []byte, receive int, retry ...Retry)
([]byte, error)
func (c *PoolConn) SendRecvPkg(data []byte, option ...PkgOption) ([]
byte, error)
func (c *PoolConn) SendRecvPkgWithTimeout(data []byte, timeout time.
Duration, option ...PkgOption) ([]byte, error)
func (c *PoolConn) SendRecvWithTimeout(data []byte, receive int,
timeout time.Duration, retry ...Retry) ([]byte, error)
func (c *PoolConn) SendWithTimeout(data []byte, timeout time.Duration,
retry ...Retry) error
```

gtcp.PoolConngtcp.Conngtcp.Conn

```

package main

import (
    "fmt"
    "time"
    "github.com/gogf/gf/v2/net/gtcp"
    "github.com/gogf/gf/v2/os/glog"
    "github.com/gogf/gf/v2/os/gtime"
)

func main() {
    // Server
    go gtcp.NewServer("127.0.0.1:8999", func(conn *gtcp.Conn) {
        defer conn.Close()
        for {
            data, err := conn.Recv(-1)
            if len(data) > 0 {
                if err := conn.Send(append([]byte("> "), data...)); err !=
nil {
                    fmt.Println(err)
                }
            }
            if err != nil {
                break
            }
        }
    }).Run()

    time.Sleep(time.Second)

    // Client
    for {
        if conn, err := gtcp.NewPoolConn("127.0.0.1:8999"); err == nil {
            if b, err := conn.SendRecv([]byte(gtime.Datetime()), -1); err
== nil {
                fmt.Println(string(b), conn.LocalAddr(), conn.RemoteAddr())
            } else {
                fmt.Println(err)
            }
            conn.Close()
        } else {
            glog.Error(err)
        }
        time.Sleep(time.Second)
    }
}

```

ServergoroutineClientmain goroutineServerClient1ServerServerClient

```

> 2018-07-11 23:29:54 127.0.0.1:55876 127.0.0.1:8999
> 2018-07-11 23:29:55 127.0.0.1:55876 127.0.0.1:8999
> 2018-07-11 23:29:56 127.0.0.1:55876 127.0.0.1:8999
> 2018-07-11 23:29:57 127.0.0.1:55876 127.0.0.1:8999
> 2018-07-11 23:29:58 127.0.0.1:55876 127.0.0.1:8999
...

```

Clientgtcp.NewConn("127.0.0.1:8999")gtcp.Connconn.Close()

```

package main

import (
    "fmt"
    "time"
    "github.com/gogf/gf/v2/net/gtcp"
    "github.com/gogf/gf/v2/os/glog"
    "github.com/gogf/gf/v2/os/gtime"
)

func main() {
    // Server
    go gtcp.NewServer("127.0.0.1:8999", func(conn *gtcp.Conn) {
        defer conn.Close()
        for {
            data, err := conn.Recv(-1)
            if len(data) > 0 {
                if err := conn.Send(append([]byte("> "), data...)); err !=
nil {
                    fmt.Println(err)
                }
            }
            if err != nil {
                break
            }
            return
        }
    }).Run()

    time.Sleep(time.Second)

    // Client
    for {
        if conn, err := gtcp.NewPoolConn("127.0.0.1:8999"); err == nil {
            if b, err := conn.SendRecv([]byte(gtime.Datetime()), -1); err
== nil {
                fmt.Println(string(b), conn.LocalAddr(), conn.RemoteAddr())
            } else {
                fmt.Println(err)
            }
            conn.Close()
        } else {
            glog.Error(err)
        }
        time.Sleep(time.Second)
    }
}

```

```

> 2018-07-20 12:56:15 127.0.0.1:59368 127.0.0.1:8999
EOF
> 2018-07-20 12:56:17 127.0.0.1:59376 127.0.0.1:8999
EOF
> 2018-07-20 12:56:19 127.0.0.1:59378 127.0.0.1:8999
EOF
...

```

ServerClientIOServerServerEOFClientClosegtcp.NewPoolConnClientServerClientgtcp.Conn
IOgtcpClose(gtcp.PoolConn)gtcp.NewPoolConn