

TCP-

gtcpgtcp.Conn

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

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

```
type Conn
func NewConn(addr string, timeout ...int) (*Conn, error)
func NewConnByNetConn(conn net.Conn) *Conn
func NewConnKeyCrt(addr, crtFile, keyFile string) (*Conn, error)
func NewConnTLS(addr string, tlsConfig *tls.Config) (*Conn, error)
func (c *Conn) Close() error
func (c *Conn) LocalAddr() net.Addr
func (c *Conn) Recv(length int, retry ...Retry) ([]byte, error)
func (c *Conn) RecvLine(retry ...Retry) ([]byte, error)
func (c *Conn) RecvWithTimeout(length int, timeout time.Duration,
retry ...Retry) ([]byte, error)
func (c *Conn) RemoteAddr() net.Addr
func (c *Conn) Send(data []byte, retry ...Retry) error
func (c *Conn) SendRecv(data []byte, receive int, retry ...Retry) ([]
byte, error)
func (c *Conn) SendRecvWithTimeout(data []byte, receive int, timeout
time.Duration, retry ...Retry) ([]byte, error)
func (c *Conn) SendWithTimeout(data []byte, timeout time.Duration,
retry ...Retry) error
func (c *Conn) SetDeadline(t time.Time) error
func (c *Conn) SetRecvBufferWait(bufferWaitDuration time.Duration)
func (c *Conn) SetRecvDeadline(t time.Time) error
func (c *Conn) SetSendDeadline(t time.Time) error
```

TCPSendretrySendSendTCP Write()

TCPSendRecvgtcp.ConnSendRecv

TCPRecvretryRecvlengthlength < 0length = 0

Recv(-1)()SendPkg/RecvPkg/

gtcp.ConnTCPtimeout

gtcp.Conn

Content Menu

-
-
-
-
- 1
- 2
- 3HTTP

```

package main

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

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 {
                fmt.Println(string(data))
            }
            if err != nil {
                break
            }
        }
    }).Run()

    time.Sleep(time.Second)

    // Client
    conn, err := gtcp.NewConn("127.0.0.1:8999")
    if err != nil {
        panic(err)
    }
    for i := 0; i < 10000; i++ {
        if err := conn.Send([]byte(gconv.String(i))); err != nil {
            glog.Error(err)
        }
        time.Sleep(time.Second)
    }
}

```

1. Server
2. Client1SocketSend
3. Server

```

2018-07-11 22:11:08.650 0
2018-07-11 22:11:09.651 1
2018-07-11 22:11:10.651 2
2018-07-11 22:11:11.651 3
2018-07-11 22:11:12.651 4
2018-07-11 22:11:13.651 5
2018-07-11 22:11:14.652 6
2018-07-11 22:11:15.652 7
2018-07-11 22:11:16.652 8
2018-07-11 22:11:17.652 9
2018-07-11 22:11:18.652 10
2018-07-11 22:11:19.653 11
.

```

```

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.NewConn("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)
    }
}

```

Client1ServerServerClientServer

```

> 2018-07-19 23:25:43 127.0.0.1:34306 127.0.0.1:8999
> 2018-07-19 23:25:44 127.0.0.1:34308 127.0.0.1:8999
> 2018-07-19 23:25:45 127.0.0.1:34312 127.0.0.1:8999
> 2018-07-19 23:25:46 127.0.0.1:34314 127.0.0.1:8999

```

3HTTP

gtcpHTTPheadercontent

```

package main

import (
    "fmt"
    "bytes"
    "github.com/gogf/gf/v2/net/gtcp"
    "github.com/gogf/gf/v2/util/gconv"
)

func main() {
    conn, err := gtcp.NewConn("www.baidu.com:80")
    if err != nil {
        panic(err)
    }
    defer conn.Close()

    if err := conn.Send([]byte("GET / HTTP/1.1\r\n\r\n")); err != nil {
        panic(err)
    }

    header      := make([]byte, 0)
    content      := make([]byte, 0)
    contentLength := 0
    for {
        data, err := conn.RecvLine()
        // header
        if len(data) > 0 {
            array := bytes.Split(data, []byte(": "))
            //
            if contentLength == 0 && len(array) == 2 && bytes.EqualFold([]
byte("Content-Length"), array[0]) {
                contentLength = gconv.Int(string(array[1][:len(array[1])
-1]))
            }
            header = append(header, data...)
            header = append(header, '\n')
        }
        // header
        if contentLength > 0 && len(data) == 1 {
            content, _ = conn.Recv(contentLength)
            break
        }
        if err != nil {
            fmt.Errorf("ERROR: %s\n", err.Error())
            break
        }
    }

    if len(header) > 0 {
        fmt.Println(string(header))
    }
    if len(content) > 0 {
        fmt.Println(string(content))
    }
}

```

```
HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: no-cache
Connection: Keep-Alive
Content-Length: 14615
Content-Type: text/html
Date: Sat, 21 Jul 2018 04:21:03 GMT
Etag: "5b3c3650-3917"
Last-Modified: Wed, 04 Jul 2018 02:52:00 GMT
P3p: CP=" OTI DSP COR IVA OUR IND COM "
Pragma: no-cache
Server: BWS/1.1
...
()
```