

# TCP-TLS

gtcpTLSTLSNewServerTLSNewServerKeyCrtTLSTLSNewConnKeyCrtNewConnTLS

<https://github.com/gogf/gf/v2/tree/master/example/net/gtcp/tls>

```
package main

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

func main() {
    address := "127.0.0.1:8999"
    crtFile := "server.crt"
    keyFile := "server.key"
    // TLS Server
    go gtcp.NewServerKeyCrt(address, crtFile, keyFile, func(conn *gtcp.Conn) {
        defer conn.Close()
        for {
            data, err := conn.Recv(-1)
            if len(data) > 0 {
                fmt.Println(string(data))
            }
            if err != nil {
                // if client closes, err will be: EOF
                g.Log().Error(err)
                break
            }
        }
    }).Run()

    time.Sleep(time.Second)

    // Client
    conn, err := gtcp.NewConnKeyCrt(address, crtFile, keyFile)
    if err != nil {
        panic(err)
    }
    defer conn.Close()
    for i := 0; i < 10; i++ {
        if err := conn.Send([]byte(gconv.String(i))); err != nil {
            g.Log().Error(err)
        }
        time.Sleep(time.Second)
        if i == 5 {
            conn.Close()
            break
        }
    }

    // exit after 5 seconds
    time.Sleep(5 * time.Second)
}
```

panic: x509: certificate has expired or is not yet valid

```

package main

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

func main() {
    address := "127.0.0.1:8999"
    crtFile := "server.crt"
    keyFile := "server.key"
    // TLS Server
    go gtcp.NewServerKeyCrt(address, crtFile, keyFile, func(conn *gtcp.Conn) {
        defer conn.Close()
        for {
            data, err := conn.Recv(-1)
            if len(data) > 0 {
                fmt.Println(string(data))
            }
            if err != nil {
                // if client closes, err will be: EOF
                g.Log().Error(err)
                break
            }
        }
    }).Run()

    time.Sleep(time.Second)

    // Client
    tlsConfig, err := gtcp.LoadKeyCrt(crtFile, keyFile)
    if err != nil {
        panic(err)
    }
    tlsConfig.InsecureSkipVerify = true

    conn, err := gtcp.NewConnTLS(address, tlsConfig)
    if err != nil {
        panic(err)
    }
    defer conn.Close()
    for i := 0; i < 10; i++ {
        if err := conn.Send([]byte(gconv.String(i))); err != nil {
            g.Log().Error(err)
        }
        time.Sleep(time.Second)
        if i == 5 {
            conn.Close()
            break
        }
    }

    // exit after 5 seconds
    time.Sleep(5 * time.Second)
}

```

```
0
1
2
3
4
5
2019-06-05 00:13:12.488 [ERRO] EOF
Stack:
1. /Users/john/Workspace/Go/GOPATH/src/github.com/gogf/gf/v2/geg/net/gtcp/tls/gtcp_server_client.go:25
```

5EOF