

WebSocket

goframewebsOCKETechogoframewebsOCKETHTML5

HTML5

H5

```
<!DOCTYPE html>
<html lang="zh">
<head>
  <title>gf websocket echo server</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
  <link rel="stylesheet" href="//cdn.bootcss.com/bootstrap/3.3.5/css
/bootstrap.min.css">
  <script src="//cdn.bootcss.com/jquery/1.11.3/jquery.min.js"></script>
</head>
<body>
<div class="container">
  <div class="list-group" id="divShow"></div>
  <div>
    <div><input class="form-control" id="txtContent" autofocus
placeholder=""></div>
    <div><button class="btn btn-default" id="btnSend" style="margin-
top:15px"> </button></div>
  </div>
</div>
</body>
</html>

<script type="application/javascript">
  //
  function showInfo(content) {
    $("<div class=\"list-group-item list-group-item-info\">" + content
+ "</div>").appendTo("#divShow")
  }
  //
  function showWarning(content) {
    $("<div class=\"list-group-item list-group-item-warning\">" +
content + "</div>").appendTo("#divShow")
  }
  //
  function showSuccess(content) {
    $("<div class=\"list-group-item list-group-item-success\">" +
content + "</div>").appendTo("#divShow")
  }
  //
  function showError(content) {
    $("<div class=\"list-group-item list-group-item-danger\">" +
content + "</div>").appendTo("#divShow")
  }

  $(function () {
    const url = "ws://127.0.0.1:8199/ws";
    let ws = new WebSocket(url);
    try {
      // ws
      ws.onopen = function () {
        showInfo("WebSocket Server [" + url + "] ");
      };
      // ws
      ws.onclose = function () {
        if (ws) {
          ws.close();
          ws = null;
        }
        showError("WebSocket Server [" + url + "] ");
      };
    }
  });
}
```

Content Menu

- [HTML5](#)
- [WebSocket](#)
- [HTTPSWebSocket](#)
- [Websocket](#)
- [WebSocket Client](#)

```

    };
    // ws
    ws.onerror = function () {
        if (ws) {
            ws.close();
            ws = null;
        }
        showError("WebSocket Server [" + url + "] ");
    };
    // ws
    ws.onmessage = function (result) {
        showWaring(" > " + result.data);
    };
} catch (e) {
    alert(e.message);
}

//
$("#btnSend").on("click", function () {
    if (ws == null) {
        showError("WebSocket Server [" + url + "] F5!");
        return;
    }
    const content = $.trim($("#txtContent").val()).replace("/[\n]
/g", "");
    if (content.length <= 0) {
        alert("!");
        return;
    }
    $("#txtContent").val("")
    showSuccess(content);
    ws.send(content);
});

//
$("#txtContent").on("keydown", function (event) {
    if (event.keyCode === 13) {
        $("#btnSend").trigger("click");
    }
});
})

</script>

```

ws://127.0.0.1:8199/ws

- websocket
- websocket
- websocket

WebSocket

```

package main

import (
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/net/ghttp"
    "github.com/gogf/gf/v2/os/gctx"
    "github.com/gogf/gf/v2/os/gfile"
    "github.com/gogf/gf/v2/os/glog"
)

var ctx = gctx.New()

func main() {
    s := g.Server()
    s.BindHandler("/ws", func(r *ghttp.Request) {
        ws, err := r.WebSocket()
        if err != nil {
            glog.Error(ctx, err)
            r.Exit()
        }
        for {
            msgType, msg, err := ws.ReadMessage()
            if err != nil {
                return
            }
            if err = ws.WriteMessage(msgType, msg); err != nil {
                return
            }
        }
    })
    s.SetServerRoot(gfile.MainPkgPath())
    s.SetPort(8199)
    s.Run()
}

```

1. WebSocket

```

websockethttpghttp.Request.WebSocketr.WebSocket()websocketWebSocketwebs
ocketwebsocketerror

```

1. ReadMessage & WriteMessage

```

websocket(ReadMessage & WriteMessage)msgTypemsgTypemsgType

```

HTTPSWebSocket

```

HTTPSWebSocketWebServerHTTPSWebSocket wss://HTML5WebSocketwss://127.0.0.1:8199
/ws

```

```

package main

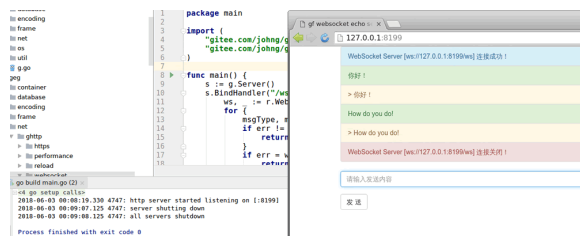
import (
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/net/ghttp"
    "github.com/gogf/gf/v2/os/gctx"
    "github.com/gogf/gf/v2/os/gfile"
    "github.com/gogf/gf/v2/os/glog"
)

var ctx = gctx.New()

func main() {
    s := g.Server()
    s.BindHandler("/wss", func(r *ghttp.Request) {
        ws, err := r.WebSocket()
        if err != nil {
            glog.Error(ctx, err)
            r.Exit()
        }
        for {
            msgType, msg, err := ws.ReadMessage()
            if err != nil {
                return
            }
            if err = ws.WriteMessage(msgType, msg); err != nil {
                return
            }
        }
    })
    s.SetServerRoot(gfile.MainPkgPath())
    s.EnableHTTPS("../https/server.crt", "../https/server.key")
    s.SetPort(8199)
    s.Run()
}

```

main.go <http://127.0.0.1:8199/websocket>



Websocket

GoFramewebsocket(origin)websocket

1. origin: r.WebSocket()origin()r.Exit()
2. websocket:

WebSocket Client

```
package main

import (
    "crypto/tls"
    "fmt"
    "net/http"
    "time"

    "github.com/gogf/gf/v2/net/gclient"
    "github.com/gorilla/websocket"
)

func main() {
    client := gclient.NewWebSocket()
    client.HandshakeTimeout = time.Second //
    client.Proxy = http.ProxyFromEnvironment //
    client.TLSClientConfig = &tls.Config{} //  tls

    conn, _, err := client.Dial("ws://127.0.0.1:8199/ws", nil)
    if err != nil {
        panic(err)
    }
    defer conn.Close()

    err = conn.WriteMessage(websocket.TextMessage, []byte("hello
word"))
    if err != nil {
        panic(err)
    }

    mt, data, err := conn.ReadMessage()
    if err != nil {
        panic(err)
    }
    fmt.Println(mt, string(data))
}
```