

Go Web Examples Courses Easy to follow Video Courses
about Go. Now in early access!

Learn more → ×

Websockets

Hey, Philipp here!

I'd like to tell you, that my platform **Go Web Examples Courses** just launched. Enjoy easy to follow video courses about web development in Go. Make sure to check out the special offer I have for early supporters.

We'll see us over there! :)

Learn more



Websockets

This example will show how to work with websockets in Go. We will build a simple server which echoes back everything we send to it. For this we have to [go get](#) the popular gorilla/websocket library like so:

```
$ go get github.com/gorilla/websocket
```

From now on, every application we write will be able to make use of this library.

```
// websockets.go
package main

import (
    "fmt"
    "net/http"
```

```

    "github.com/gorilla/websocket"
)

var upgrader = websocket.Upgrader{
    ReadBufferSize: 1024,
    WriteBufferSize: 1024,
}

func main() {
    http.HandleFunc("/echo", func(w http.ResponseWriter, r *http.Request) {
        conn, _ := upgrader.Upgrade(w, r, nil) // error ignored for sake of simplicity

        for {
            // Read message from browser
            msgType, msg, err := conn.ReadMessage()
            if err != nil {
                return
            }

            // Print the message to the console
            fmt.Printf("%s sent: %s\n", conn.RemoteAddr(), string(msg))

            // Write message back to browser
            if err = conn.WriteMessage(msgType, msg); err != nil {
                return
            }
        }
    })

    http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        http.ServeFile(w, r, "websockets.html")
    })

    http.ListenAndServe(":8080", nil)
}

```

```

<!-- websockets.html -->
<input id="input" type="text" />
<button onclick="send()">Send</button>
<pre id="output"></pre>
<script>
    var input = document.getElementById("input");

```

```
var output = document.getElementById("output");
var socket = new WebSocket("ws://localhost:8080/echo");

socket.onopen = function () {
    output.innerHTML += "Status: Connected\n";
};

socket.onmessage = function (e) {
    output.innerHTML += "Server: " + e.data + "\n";
};

function send() {
    socket.send(input.value);
    input.value = "";
}
</script>
```

```
$ go run websockets.go
[127.0.0.1]:53403 sent: Hello Go Web Examples, you're doing great!
```

Send

Status: Connected

Server: Hello Go Web Examples, you're doing great!

[Legal Disclosure](#) [Privacy Statement](#)