



OST

Eastern Switzerland
University of Applied Sciences

Distributed Systems (DSy)

Web Sockets

Thomas Bocek

10.03.2022

Learning Goals

- Lecture 3 (Tor)
 - What is Tor?
 - How does it work?
 - Why do we need onion/hidden services?
 - How to setup an onion service?
- Lecture 3 (Web Sockets)
 - What is a web socket?
 - How can I implement it?

WebSockets

- Full-duplex communication over TCP [[overview](#)]
 - REST / JSON is in one direction
- How can the server notify the browser (client?)
 - **Polling**
 - Short: request e.g. every 0.5s
 - Long: request until timeout or reply
 - Server Sent Events (**alternative**) **SSE**
 - One way communication from server to browser (client)
 - Server receives a regular HTTP request, keeps connection open, server can now push data to the client
 - **WebSockets**

- HTTP handshake, then upgrade to communication channel

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
Origin: http://example.com
```

Server response:

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: H5m1rc0sM1YukAGmm50PpG2HaGwk=
Sec-WebSocket-Protocol: chat
```

- Data can be text or binary
- With SSL/TLS → wss://
 - Some configuration required on LBs / RRs

WebSockets

- GoLang / HTML/JS Example
 - <https://github.com/tbocek/DSy>
 - Any language supports WebSockets

```
<script type="text/javascript">
  const connection = new WebSocket('ws://localhost:8080/ws');
  connection.onopen = function () {
    connection.send('start');
  };
  let counter = 0;
  connection.onmessage = function (e) {
    console.log('update websocket: ' + counter++);
    const date = JSON.parse(e.data);
    const element = document.getElementById('text');
    element.innerHTML = 'Time: ' + date.now;
  };
</script>
```

```
go func(conn *websocket.Conn) {
    for {
        w, err := conn.NextWriter(messageType)
        if err != nil {
            log.Printf("NextWriter error: %v", err)
            conn.Close()
            return
        }

        json, err := json.Marshal(Date{
            Now: time.Now().Format(time.RFC3339),
        })
        if err != nil {
            log.Printf("cannot marshal json: %v", err)
            conn.Close()
            return
        }

        w.Write(json)
        w.Close()
        time.Sleep(1 * time.Second)
    }
}(conn)
```