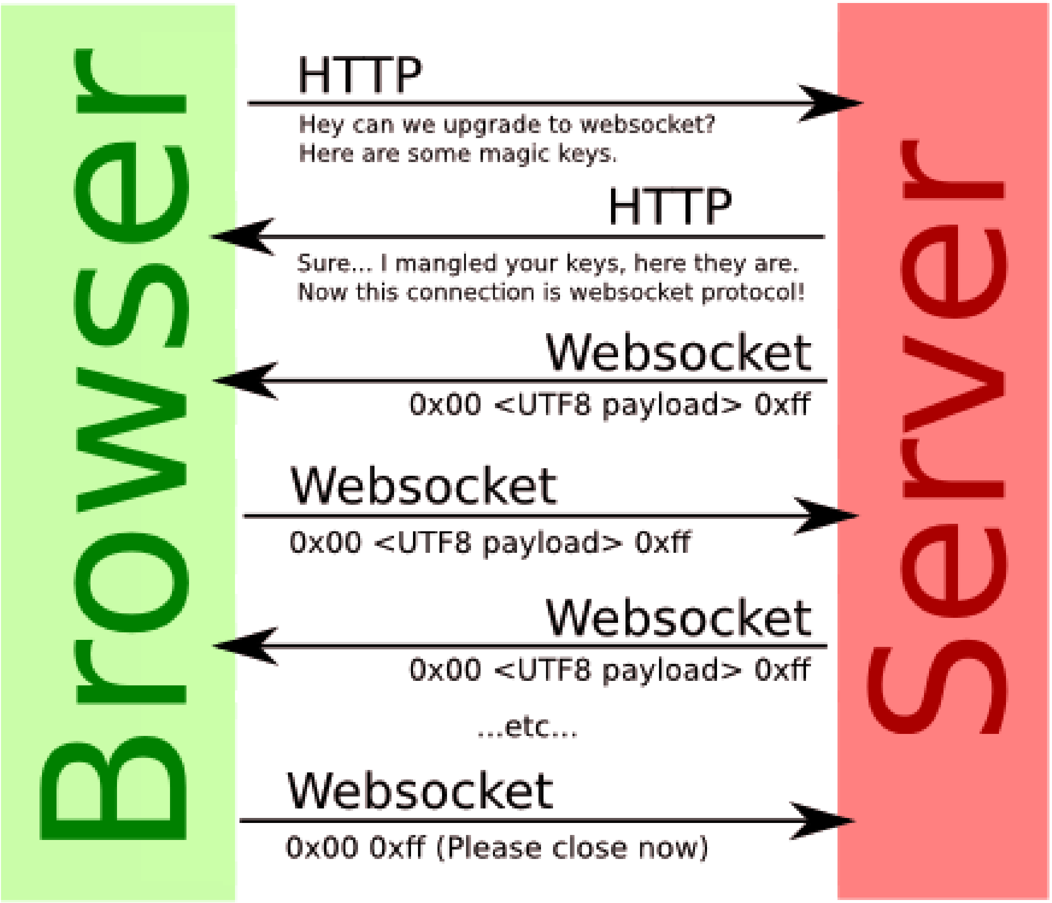
SOCKETS

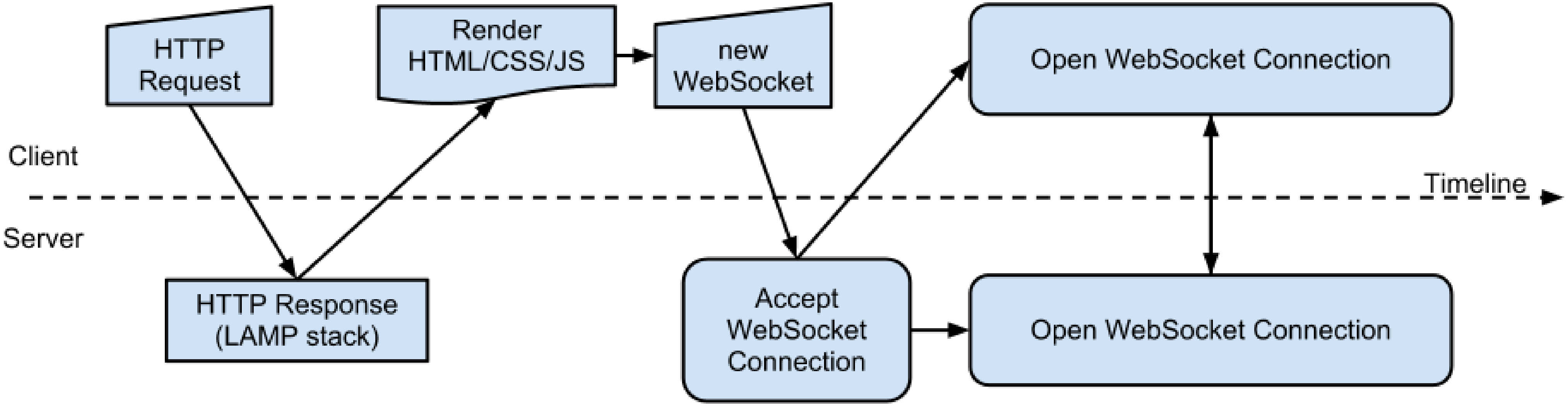
Andrew Sheehan
MET CS602

WHAT IS A SOCKET?

A bidirectional, fully duplexed channel between clients and your server.

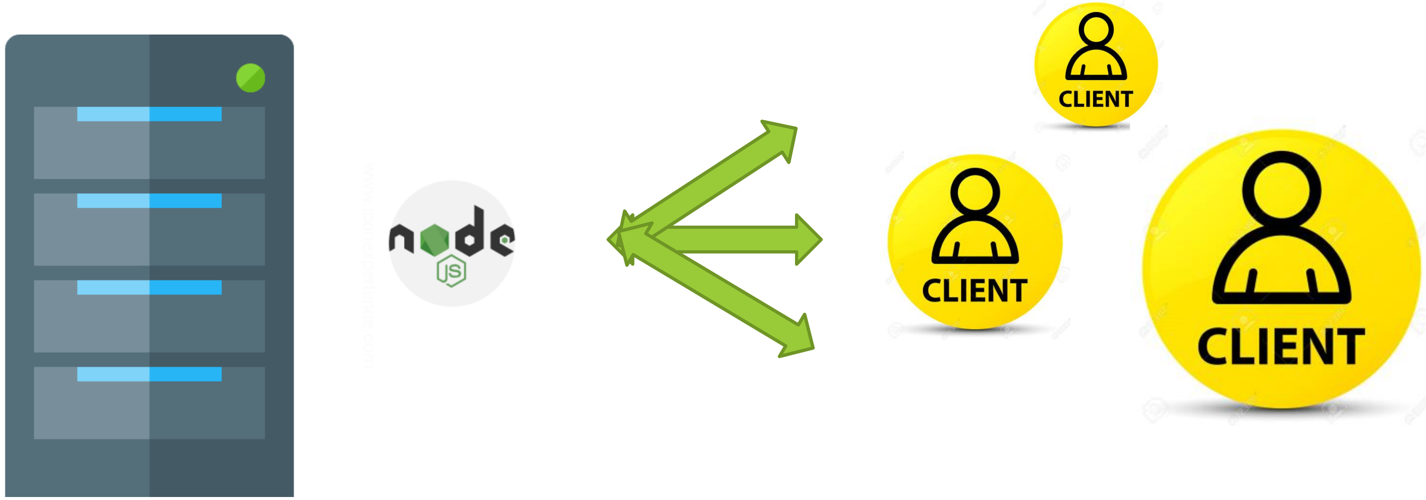

Browser
Server
HTTP
Hey can we upgrade to websocket?
Here are some magic keys.
HTTP
Sure... I mangled your keys, here they are.
Now this connection is websocket protocol!
Websocket
0x00 <UTF8 payload> 0xff
Websocket
0x00 <UTF8 payload> 0xff
Websocket
0x00 <UTF8 payload> 0xff
...etc...
Websocket
0x00 0xff (Please close now)

STANDARD APPROACH “WHAT HAPPENS”

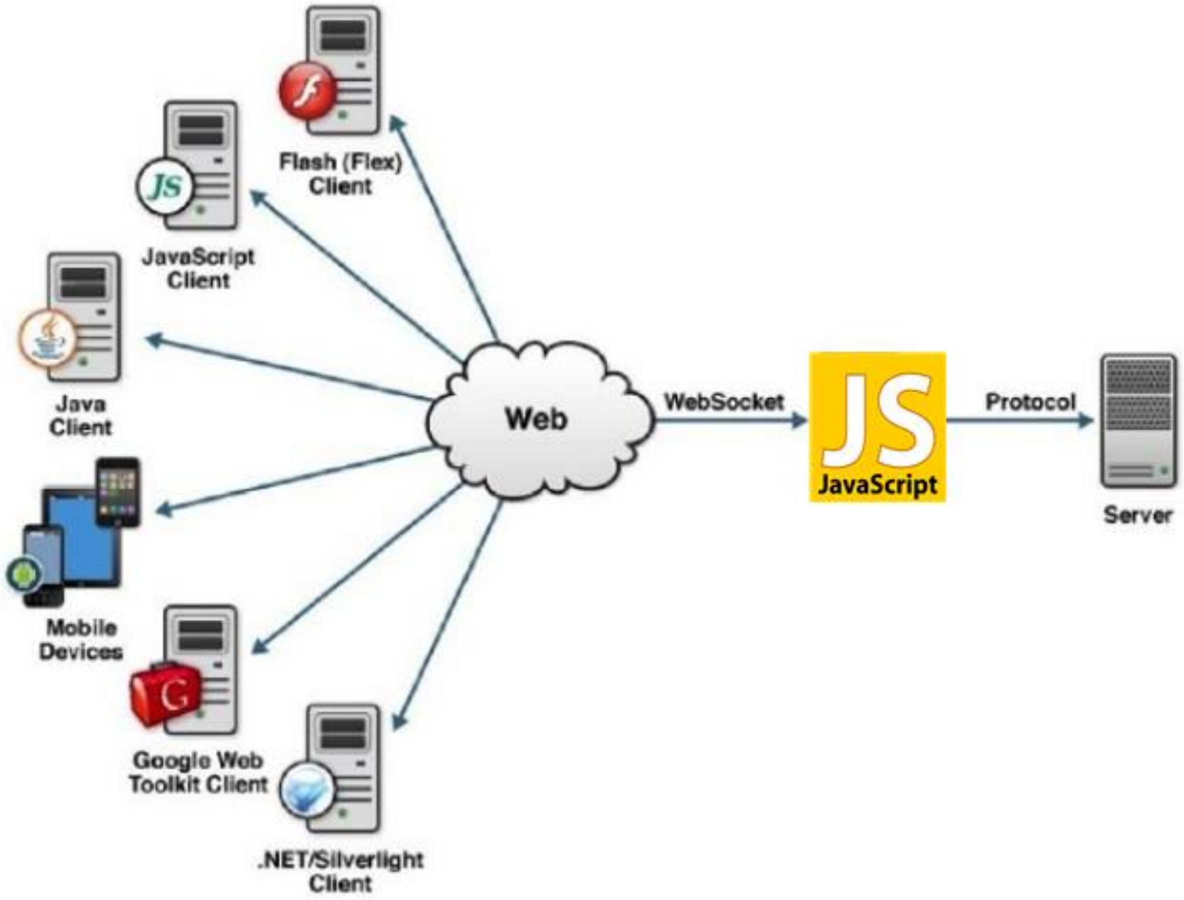

HTTP Request
Render HTML/CSS/JS
new WebSocket
Open WebSocket Connection
Client
Timeline
Server
HTTP Response (LAMP stack)
Accept WebSocket Connection
Open WebSocket Connection

SOCKETS
PUSHING MESSAGES

Having the ability to push messages out to clients means no polling (less hits on your server)



JAVASCRIPT WEBSOCKET



The WebSocket Protocol is a widely supported open standard for developing real-time applications

ONCE CONNECTED
STAYS CONNECTED

Once a WebSocket connection is established the connection stays open until the client or server decides to close this connection

WEBSOCKET
COMMON PROTOCOLS

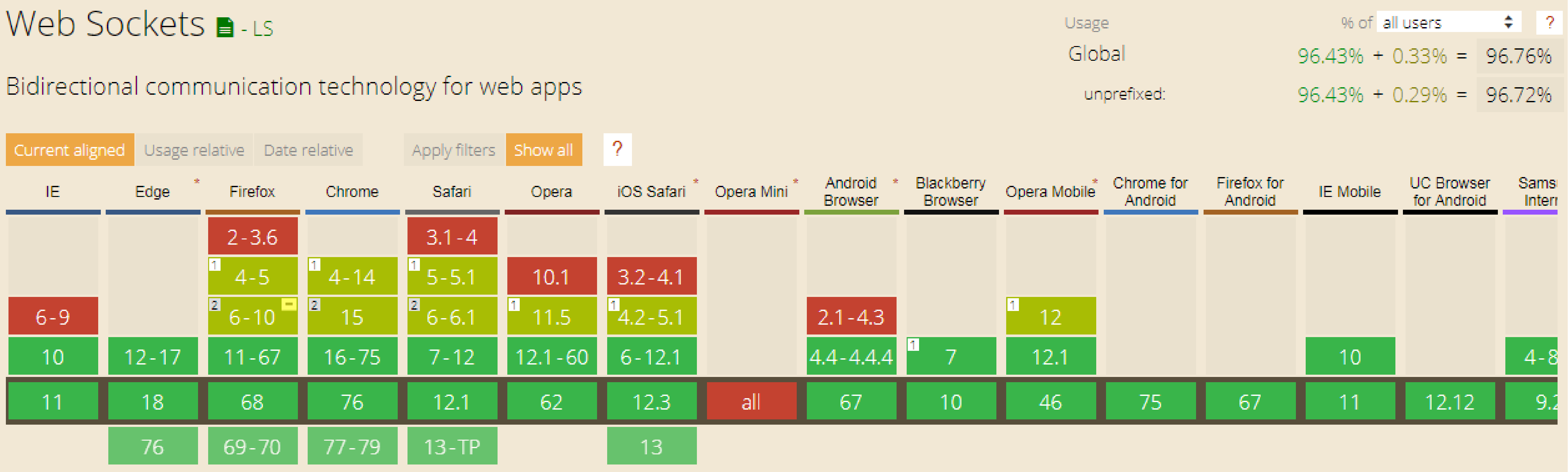
IPC

Inter-Process Communication allows processes to use shared data (like chats, SMS, messages..)

TCP

Transmission Control Protocol One of the original protocols when the ‘network’ started.

CANIUSE
YES



Web Sockets - LS

Bidirectional communication technology for web apps

Usage % of all users ?

Global 96.43% + 0.33% = 96.76%

unprefixed: 96.43% + 0.29% = 96.72%

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android	IE Mobile	UC Browser for Android	Sams Interr
		2-3.6		3.1-4											
		[1] 4-5	[1] 4-14	[1] 5-5.1	10.1	3.2-4.1									
6-9		[2] 6-10	[2] 15	[2] 6-6.1	[1] 11.5	[1] 4.2-5.1		2.1-4.3		[1] 12					
10	12-17	11-67	16-75	7-12	12.1-60	6-12.1		4.4-4.4.4	[1] 7	12.1			10		4-8
11	18	68	76	12.1	62	12.3	all	67	10	46	75	67	11	12.12	9.2
	76	69-70	77-79	13-TP		13									

SOCKET.IO
WEBSOCKET API (CLIENT SIDE)

```
<script
  src=“https://cdnjs.cloudflare.com/ajax/libs/socket.io/2.2.0/socket.io.js”>
</script>
```

```
<script>
    const socket = io();
</script>
```

SOCKET.IO
WEBSOCKET API (CLIENT SIDE)

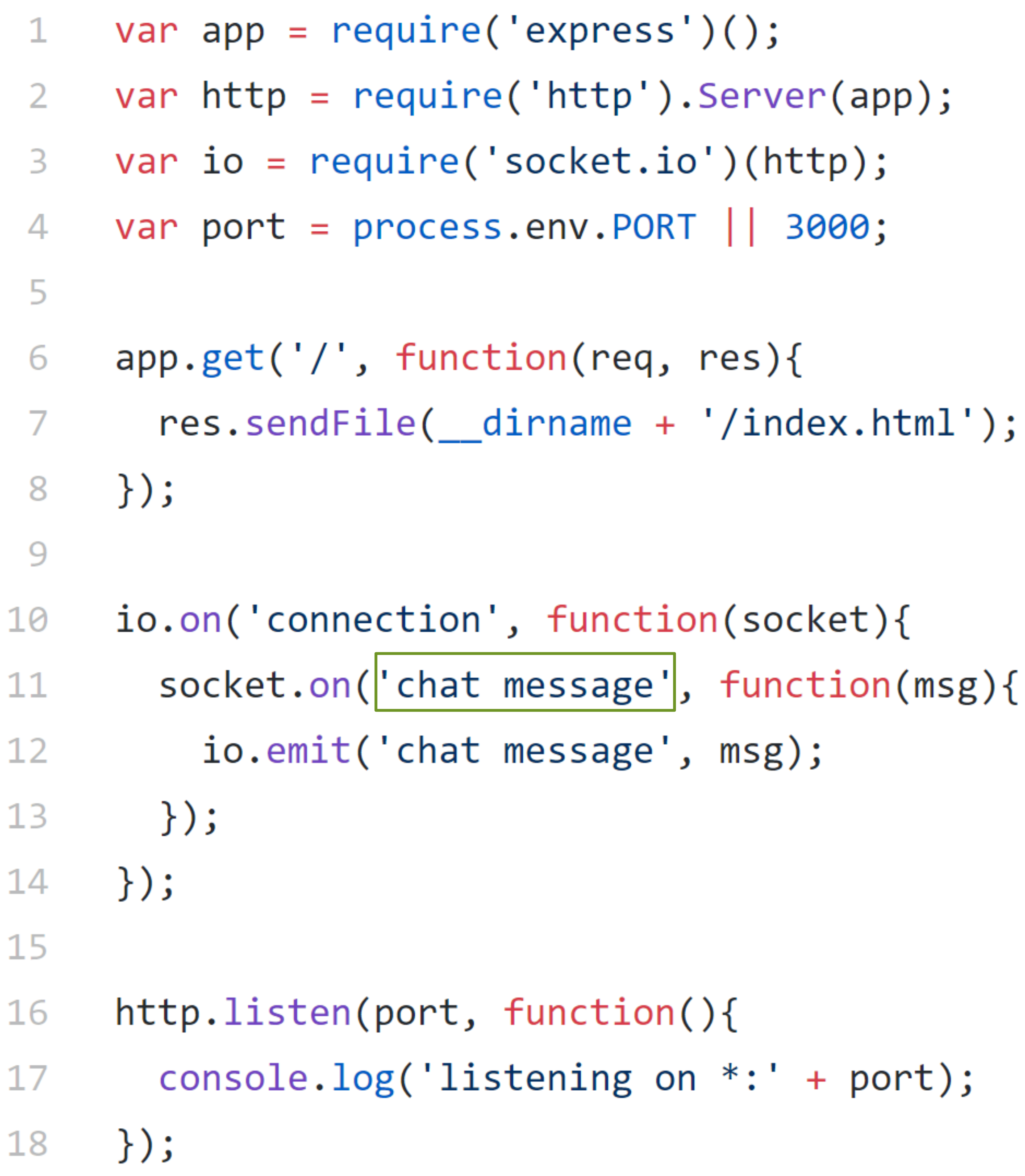
```
<script>
  const socket = io();

  // some event is triggered
  socket.emit(“chat message”, “”);
</script>
```

Has to be this value (*the event name*) in the emit() call

SOCKET.IO
MORE EXAMPLE CLIENT SIDE

```
<script>
   const socket = new WebSocket(http://localhost:3000/chat);
    // on successful connection to server
    socket.onopen( event => { socket.send("Sending message to
the server…");});
   // when receiving a msg from server
   socket.onmessage( event => { console.log(event.data); });
</script>
```

SOCKET.IO
SERVER SIDE



```
var app = require('express')();
var http = require('http').Server(app);
var io = require('socket.io')(http);
var port = process.env.PORT || 3000;

app.get('/', function(req, res){
  res.sendFile(__dirname + '/index.html');
});

io.on('connection', function(socket){
  socket.on('chat message', function(msg){
    io.emit('chat message', msg);
  });
});

http.listen(port, function(){
  console.log('listening on *:' + port);
});
```

LABORATORY

WEBSOCKET WITH SOCKET.IO