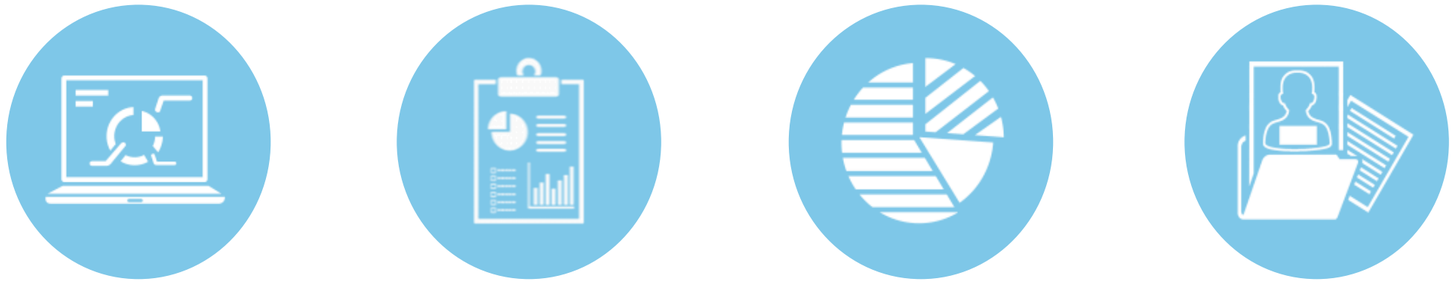
Node.js

Lesson 11—Working with Socket.IO



Lesson Overview

In this lesson, you will be able to understand and install Socket.IOand send and receive messages from Client to server.

Learning Objectives

- What is Socket.IO?
- Installing Socket.IO on Node.js
- Sending and Receiving messages from Client
- Detecting disconnects from Server and Client

Working with Socket.IO

Topic 1—What is Socket.IO?

What Is Socket.IO?

The main purpose of Socket.IO is to make real-time applications possible in all the browsers and mobile devices, blurring the differences between the different transport mechanisms.

It is easy and works real-time in JavaScript.

Socket.IO helps the developer to create bidirectional event-based, real-time communication.

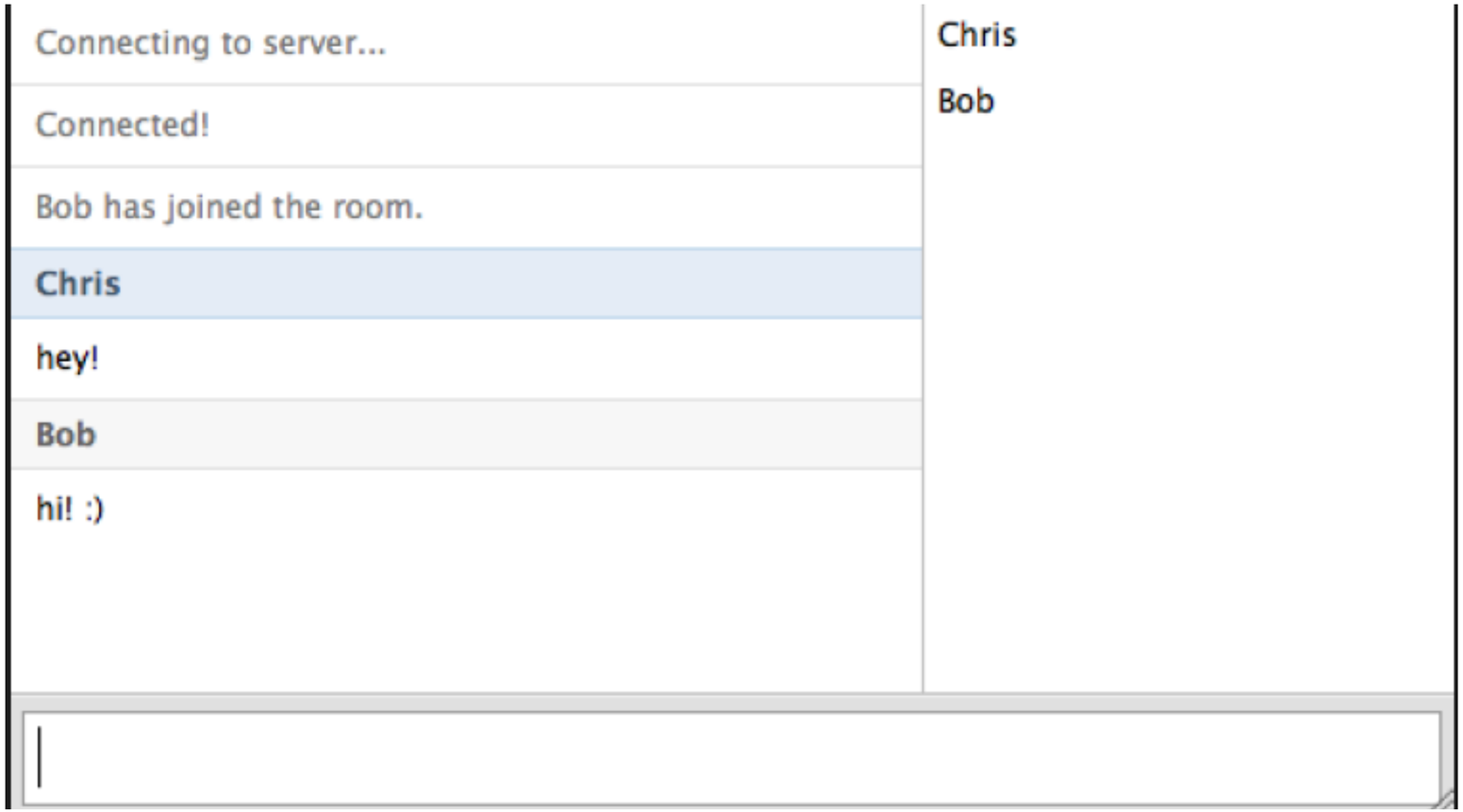
It is a platform as well as a browser which is device independent and keeps the focus equally on simplicity, speed, and reliability.

Socket.IO is built on top of the Node.js and WebSockets API.

What Is Socket.IO?

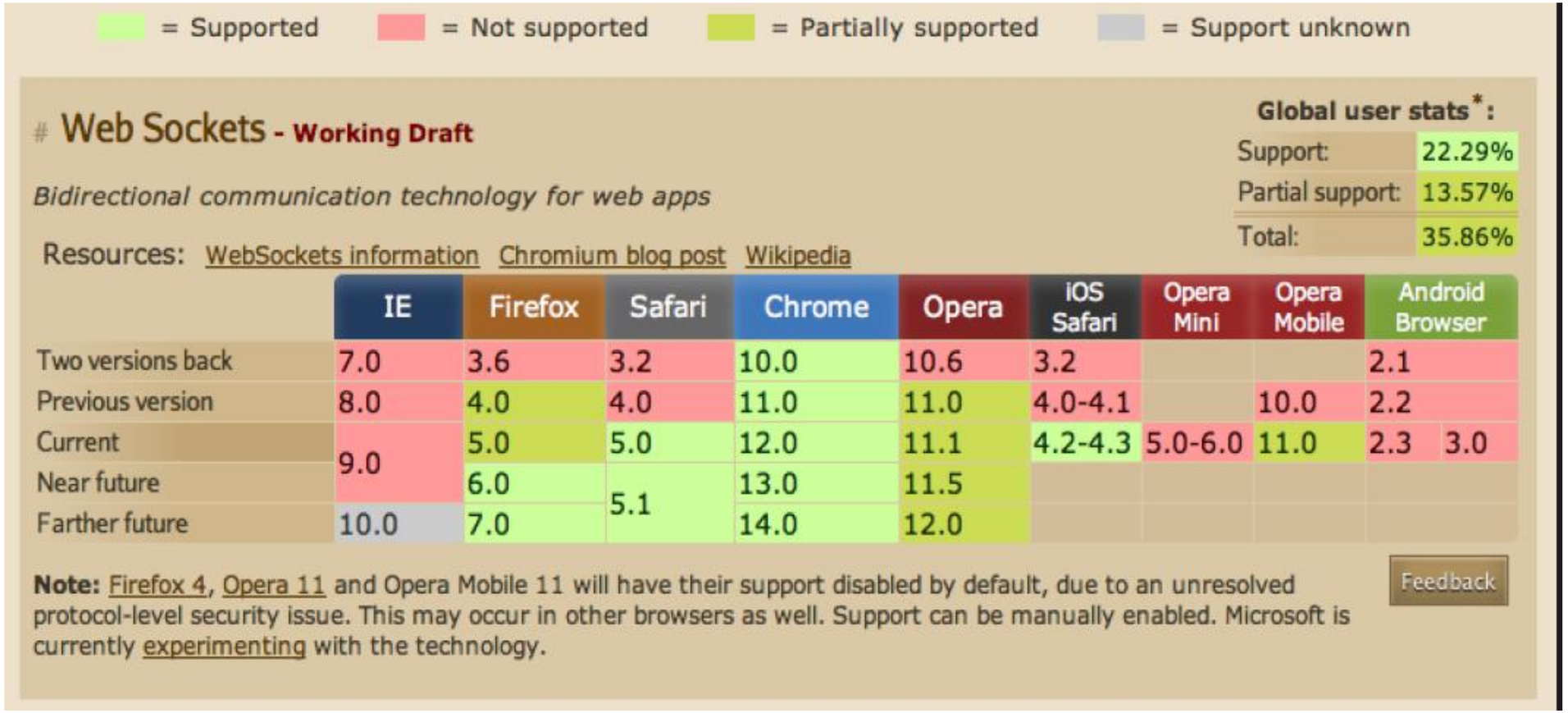
OBJECTIVE

To create an application which you can use in web-based environment



What Is Socket.IO?

WORKING CHALLENGE WITH WEBSOCKETS



= Supported = Not supported = Partially supported = Support unknown

Web Sockets - Working Draft

Bidirectional communication technology for web apps

Resources: WebSockets information Chromium blog post Wikipedia

Global user stats*:

Support:	22.29%
Partial support:	13.57%
Total:	35.86%

	IE	Firefox	Safari	Chrome	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser	
Two versions back	7.0	3.6	3.2	10.0	10.6	3.2			2.1	
Previous version	8.0	4.0	4.0	11.0	11.0	4.0-4.1		10.0	2.2	
Current	9.0	5.0	5.0	12.0	11.1	4.2-4.3	5.0-6.0	11.0	2.3	3.0
Near future		6.0	5.1	13.0	11.5					
Farther future	10.0	7.0		14.0	12.0					

Note: Firefox 4, Opera 11 and Opera Mobile 11 will have their support disabled by default, due to an unresolved protocol-level security issue. This may occur in other browsers as well. Support can be manually enabled. Microsoft is currently experimenting with the technology.

Feedback

Working with Socket.IO

Topic 2—Installing Socket.IO on Node.js

Installing Socket.IO on Node.js

- Public URL : Socket.IO: http://socket.io

- Install Socket.IO using npm:

```
npm install socket.io
```

- Documentation: https://socket.io/docs/

Installing Socket.IO on Node.js

FIRST SOCKET.IO SERVER

 # Using Express as http server:

```
var app = require('express').createServer();
var io = require('socket.io').listen(app);
io.sockets.on('connection', function (socket) {
console.log('User is connected!');
});app.listen(8080);
```

 # Using Node.js as http server:

```
var app =
require('http').createServer(callback);
var io = require('socket.io').listen(app);
app.listen(8080);
```

Installing Socket.IO on Node.js

CLIENT SIDE JS CODE

 # Client Side JS Code:

```
<script src='/socket.io/socket.io.js'></script>
<script>
var socket = io.connect();
socket.on('connect', function () {
console.log('User is connected on client APP!');
});
</script>
```

Demo for Installing Socket.IO

Working with Socket.IO

Topic 3—Sending and receiving messages from Client to Server

Sending Messages from Client to Server

You can send the messages from Client to Server using:

```
<script src='/socket.io/socket.io.js'></script>
<script>
var socket = io.connect();
socket.on('connect', function () {
console.log('User is connected!');
var Uname = prompt('What is your name');
this.emit('set username', Uname);
});
</script>
```

Demo for Sending Messages from Client to Server

Receiving Messages from Client to Server

You can receive the messages from Client to Server using:

```
var app = require('express').createServer();
var io = require('socket.io').listen(app);
io.sockets.on('connection', function (socket) {
console.log('Someone connected!');
socket.on('set username', function (username) {
socket.username = username;
console.log(username + ' just connected!');
});});
app.listen(8080);
```

Receiving Messages from Client to Server

MESSAGE ACKNOWLEDGEMENT—CLIENT SIDE

```
<script src='/socket.io/socket.io.js'></script>
<script>
var socket = io.connect();
socket.on('connect', function () {
console.log('We are connected!');
var name = prompt('your name');
this.emit('name', name, function (success) {
console.log('The server received the message!');
if (!success) {
console.log(‘name in use!');
}}); });
</script>
```

Demo for Receiving Messages from Client to Server

Working with Socket.IO

Topic 4—Detecting disconnects from Server and Client

Detecting Disconnects from Server and Client

You can detect the disconnect from Server and Client using:

```
var app = require('express').createServer();
var io = require('socket.io').listen(app);
var users = [];
io.sockets.on('connection', function (socket) {
// ...
socket.on('disconnect', function () {
var name = socket.name;
if (name) {
socket.broadcast.emit('user part', name);
users.splice(users.indexOf(name), 1);}});});
app.listen(8080);
```

Quiz

QUIZ 1

Which module requires to use Socket in Node?

a. HTTP

b. Express

c. Eventemitter

d. Both A & B

QUIZ 1

Which module requires to use Socket in Node?

a. HTTP

b. Express

c. Eventemitter

d. Both A & B

The correct answer is **d. Both A & B.**

HTTP and Express modules require to use Socket in Node.

QUIZ 2

___method is used to broadcast the message in socket.

a. ShareAll

b. Broadcast

c. ShareMessage

d. ConnectToAll

QUIZ 2

___method is used to broadcast the message in socket.

a. ShareAll

b. Broadcast

c. ShareMessage

d. ConnectToAll

The correct answer is **b. Broadcast.**

Broadcast method is used to broadcast the message in socket.

QUIZ 3

SocketIO can be used to implement real-time messaging.

a. True

b. False

QUIZ 3

SocketIO can be used to implement real-time messaging.

a. True

b. False

The correct answer is **a. True.**

SocketIO can be used to implement real-time messaging.

Key Takeaways

- Main purpose of Socket.IO is to make real-time applications possible in all the browsers and mobile devices, blurring the differences between the different transport mechanisms.
- You can install Socket.IO using: *npm install socket.io*
- You can send and receive messages from Client to Server using the Node.js and Socket.IO commands.
- You can detect the disconnects between Client and Server using the Node.js and Socket.IO commands.

Thank You