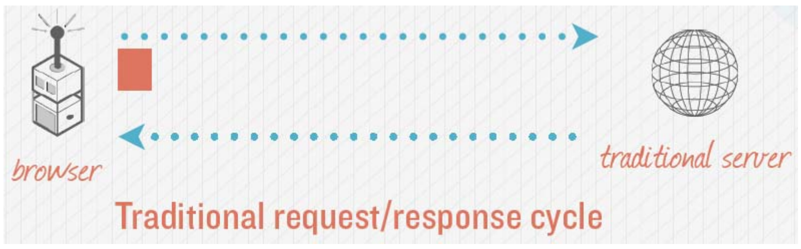
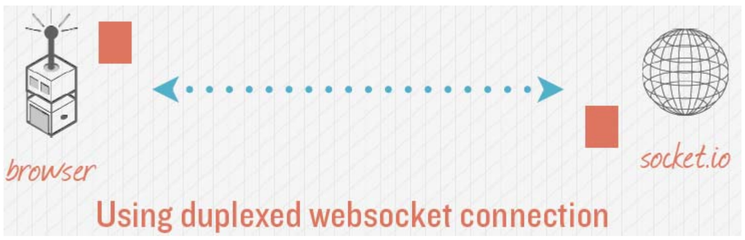
socket.io

MODULE 11

SOCKET.IO

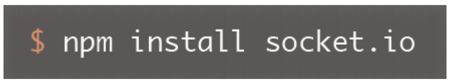
WEBSOCKETS



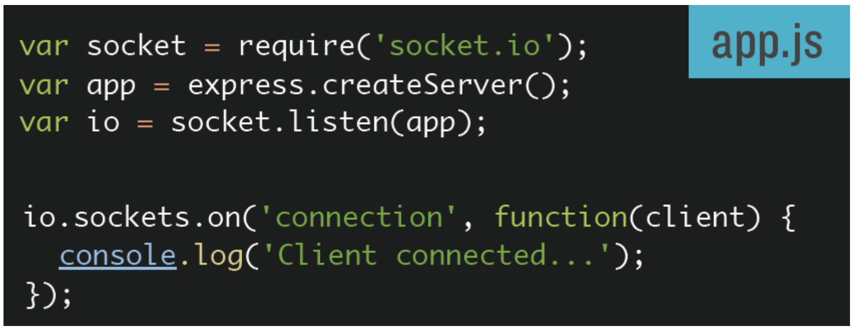


SOCKET.IO FOR WEBSOCKETS

> Abstracts websockets with fallbacks



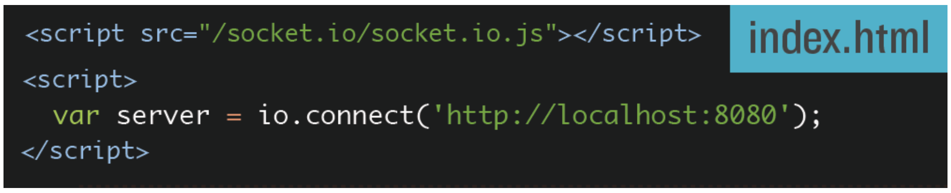
```
$ npm install socket.io
```



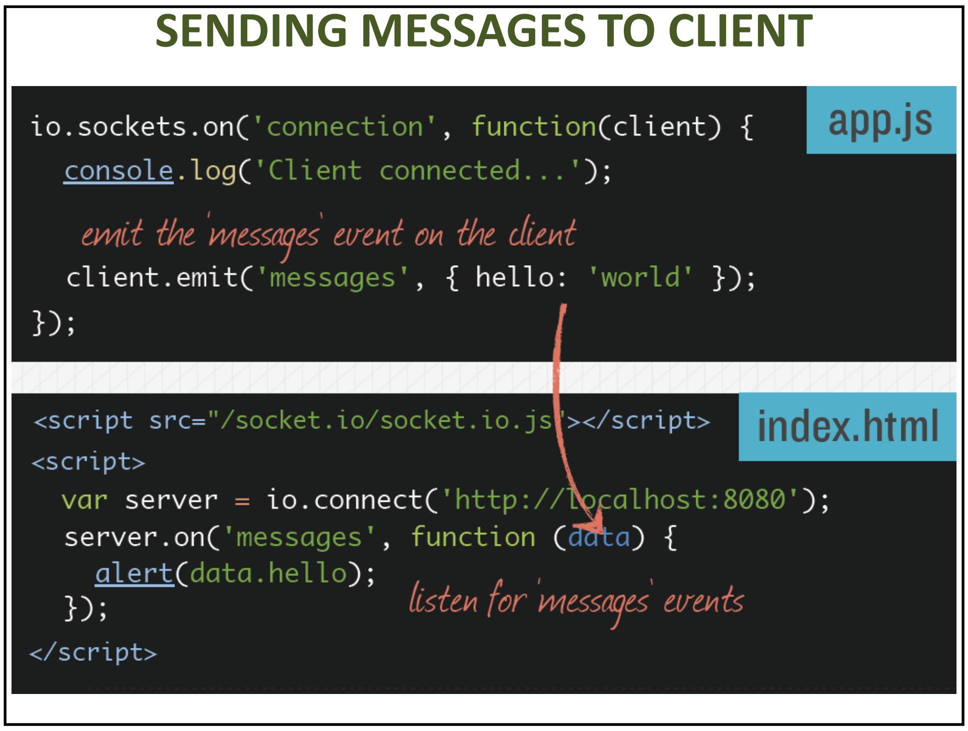
```
var socket = require('socket.io');
var app = express.createServer();
var io = socket.listen(app);

io.sockets.on('connection', function(client) {
  console.log('Client connected...');
});
```

SOCKET.IO FOR WEBSOCKETS

> Client-side



```
<script src="/socket.io/socket.io.js"></script>
<script>
  var server = io.connect('http://localhost:8080');
</script>
```



SENDING MESSAGES TO CLIENT

app.js

```
io.sockets.on('connection', function(client) {
  console.log('Client connected...');
```

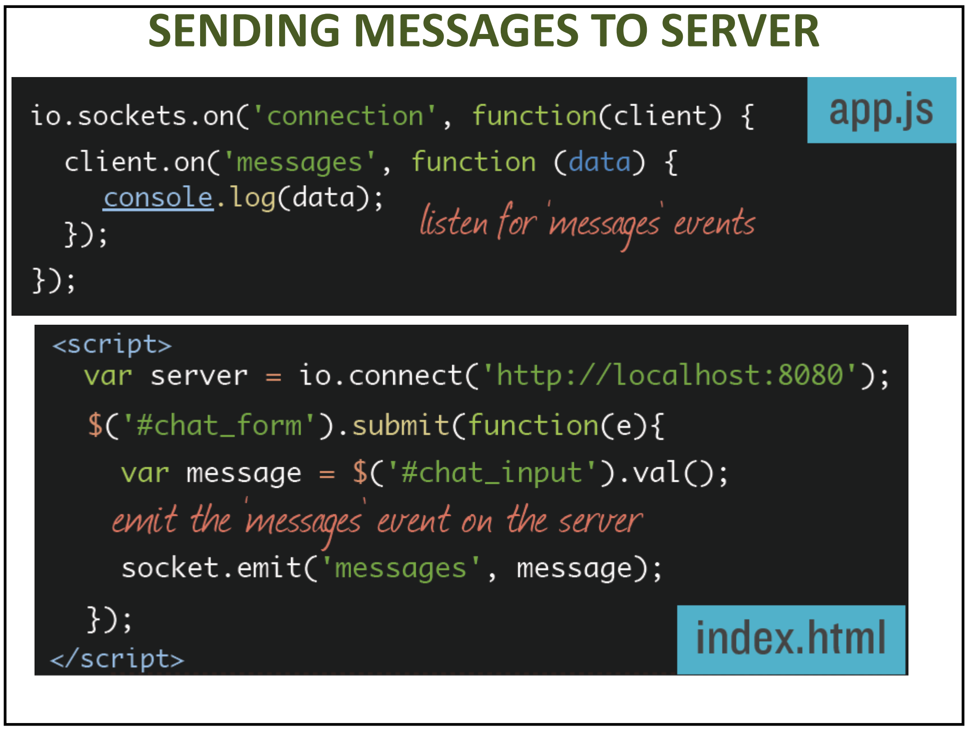
emit the 'messages' event on the client

```
  client.emit('messages', { hello: 'world' });
});
```

index.html

```
<script src="/socket.io/socket.io.js"></script>
<script>
  var server = io.connect('http://localhost:8080');
  server.on('messages', function (data) {
    alert(data.hello);
  });
</script>
```

listen for 'messages' events



SENDING MESSAGES TO SERVER

app.js

```
io.sockets.on('connection', function(client) {
  client.on('messages', function (data) {
    console.log(data);
  });
});
```

listen for 'messages' events

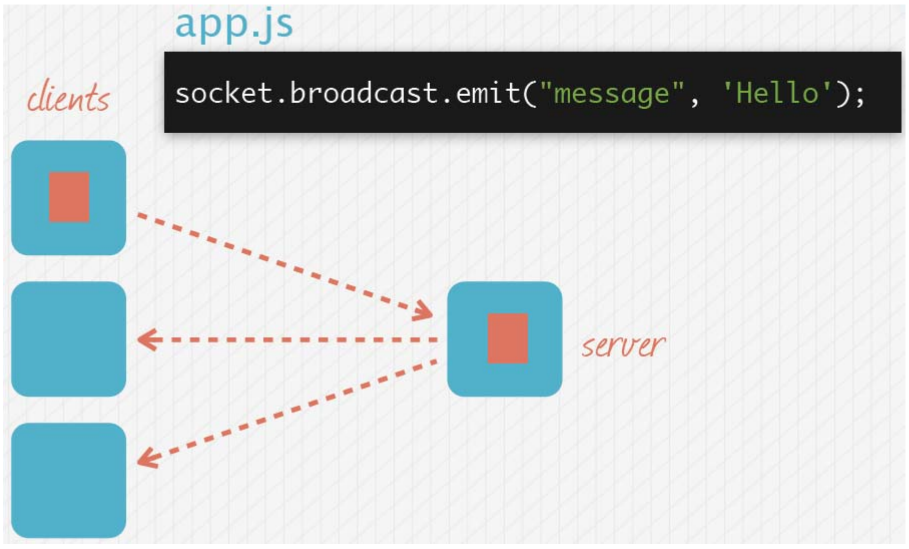
```
<script>
  var server = io.connect('http://localhost:8080');
  $('#chat_form').submit(function(e){
    var message = $('#chat_input').val();
```

emit the 'messages' event on the server

```
    socket.emit('messages', message);
  });
</script>
```

index.html

BROADCASTING MESSAGES



```
socket.broadcast.emit("message", 'Hello');
```

BROADCASTING MESSAGES

app.js

```
io.sockets.on('connection', function(client) {
  client.on('messages', function (data) {
    client.broadcast.emit("messages", data);
  });
});
```

broadcast message to all other clients connected

index.html

```
<script>
  ...
  server.on('messages', function(data) { insertMessage(data) });
</script>
```

insert message into the chat

SAVING DATA ON THE SOCKET

app.js

```
io.sockets.on('connection', function(client) {
  client.on('join', function(name) {
    client.set('nickname', name);
  });
});
```

set the nickname associated with this client

index.html

```
<script>
  var server = io.connect('http://localhost:8080');
  server.on('connect', function(data) {
    $('#status').html('Connected to chattr');
    nickname = prompt("What is your nickname?");

    server.emit('join', nickname);
  });
</script>
```

notify the server of the users nickname

SAVING DATA ON THE CLIENT

app.js

```
io.sockets.on('connection', function(client) {
  client.on('join', function(name) {
    client.set('nickname', name);
  });
  client.on('messages', function(data){
    client.get('nickname', function(err, name) {
      client.broadcast.emit("chat", name + ": " + message);
    });
  });
});
```

set the nickname associated with this client

get the nickname of this client before broadcasting message

broadcast with the name and message