

Interacting with the Web

Paul O'Fallon
@paulofallon



Outline

- Using Node as a web client
- Building a web server
- Real-time integration using Socket.IO

Making web requests in Node

```
var http = require('http');
```

Instance of `http.ClientRequest` (a `WritableStream`)

```
var req = http.request(options, function(res) {  
  // process callback  
});
```

Instance of `http.ClientResponse` (a `ReadableStream`)

- “options” can be one of the following:
 - A URL string
 - An object specifying values for host, port, method, path, headers, auth, etc.
- The returned `ClientRequest` can be written/piped to for POST requests
- The `ClientResponse` object is provided via either callback (shown above) or as a “response” event on the request object.
- `http.get()` available as a simplified interface for GET requests



Building a Web Server in Node

```
var http = require('http');
```

Instance of `http.ServerRequest` (a `ReadableStream`)

```
var server = http.createServer(function(req, res) {  
  // process request  
});  
server.listen(port, [host]);
```

Instance of `http.ServerResponse` (a `WritableStream`)

- Each request is provided via either callback (shown above) or as a “request” event on the server object
- The `ServerRequest` can be read from (or piped) for POST uploads
- The `ServerResponse` can be piped to when returning stream-oriented data in a response
- SSL support is provided by a similar `https.createServer()`



Real-Time Communication with Socket.IO



The image shows a screenshot of a web browser displaying the Socket.IO website. The browser's address bar shows 'socket.io'. The website has a dark blue background with the text 'Introducing SOCKET IO v.9' in a large, stylized font. A yellow banner with 'v.9' is next to the title. A diagonal banner in the top right corner says 'Fork me on GitHub'. Below the title, the text 'What is Socket.IO?' is followed by a paragraph: 'Socket.IO aims to make realtime apps possible in every browser and mobile device, blurring the differences between the different transport mechanisms. It's care-free realtime 100% in JavaScript.' Below this, there are two columns: 'SERVER' and 'CLIENT'. The 'SERVER' column contains a code block with JavaScript code for listening on port 80 and emitting events. The 'CLIENT' column contains a code block with JavaScript code for connecting to the server and listening for events. At the bottom, a footer reads: 'Socket.IO by Guillermo Rauch at gradebook LearnBoost Labs. Released under the MIT license - Copyright LearnBoost 2012'.

Socket.IO: the cross-browser

socket.io

Introducing
SOCKET IO v.9

Fork me on GitHub

What is Socket.IO?

Socket.IO aims to make realtime apps possible in every browser and mobile device, blurring the differences between the different transport mechanisms. It's care-free realtime 100% in JavaScript.

SERVER

```
var io = require('socket.io').listen(80);

io.sockets.on('connection', function (socket) {
  socket.emit('news', { hello: 'world' });
  socket.on('my other event', function (data) {
    console.log(data);
  });
});
```

CLIENT

```
<script src="/socket.io/socket.io.js"></script>
<script>
  var socket = io.connect('http://localhost');
  socket.on('news', function (data) {
    console.log(data);
    socket.emit('my other event', { my: 'data' });
  });
</script>
```

Socket.IO by Guillermo Rauch at gradebook LearnBoost Labs. Released under the MIT license - Copyright LearnBoost 2012

Socket.IO Exchange

Server:

```
var io = require('socket.io').listen(80);
```

```
io.sockets.on('connection', function (socket) {
```

```
  socket.emit('news', {hello: 'world' });
```

```
  socket.on('my other event', function(data) {  
    console.log(data);
```

```
  });
```

```
});
```

Browser:

```
<script src="/socket.io/socket.io.js"></script>
```

```
<script>
```

```
  var socket = io.connect('http://localhost');
```

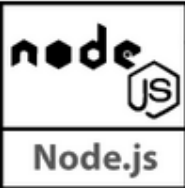
```
  socket.on('news', function (data) {  
    console.log(data);
```

```
    socket.emit('my other event', { my: 'data' });  
  });
```

```
</script>
```




To learn more about Web Apps in Node



Web Development with ExpressJS

Course on web development with express.js, a lightweight Sinatra-based web framework for node.js

10 12 16 3

Author: Hadi Hariri
Duration: 2h 13m
Level: Intermediate
Released: 9/27/2012
Features: CC

Table of ContentsDescriptionTranscriptExercise FilesAssessment

Currently using: Silverlight Player [Change]

expand allcollapse all

	Progress	Duration
Web Development with Express.js		00:28:37
An Introduction to Express.js		00:17:51
Views		00:17:48
Working with Data		00:27:14
Common and Advanced Scenarios		00:41:32

<http://www.pluralsight.com/training/Courses/TableOfContents/expressjs>

Conclusion

- Using `http.request()` as a web client
- Building a web server with `http.createServer()`
- Using `Socket.IO` for real-time communication



References

- Node.js documentation
<http://nodejs.org/api/>
- Socket.IO
<http://socket.io/>