

Introduction to Node.js

Doug Toppin

dougtoppin@gmail.com

github.com/dougtoppin/presentations

Agenda

- What is Node.js?
- npm
- Why would I use it?
- Syntax
- Some examples
- Links

What is Node.js?

- From nodejs.org
- Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

What is Node.js?

- “...JavaScript, and therefore Node.js, introduces the concept of event-driven, asynchronous callbacks, by utilizing an event loop.”
- (call a function when you're done doing something)

npm

- Node Package Manager, Node Packaged Modules (NPM)
- Installs packages and dependencies
- Without arguments discovers dependencies in current directory
- git url aware
- Packages can be: files, tarball, url
- npm search

Why would I use it?

- Really quick way to put up a web server
- Quick and easy script editing
- My experience has been that installation is much quicker/easier than Rails (and gem)

Express

- Express is a minimal and flexible node.js web application framework, providing a robust set of features for building single and multi-page, and hybrid web applications.

Syntax

Hello World

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

http.createServer(function (request, response) {
  response.writeHead(200, {'Content-Type': 'text/plain'});
  response.end('Hello World\n');
}).listen(8124);

console.log('Server running at http://127.0.0.1:8124/');
```

```
$ node example.js
Server running at http://127.0.0.1:8124/
```

```
$ curl http://127.0.0.1:8124/
Hello World
```


Syntax

```
$ cat functions.js
```

```
function say(word) {  
    console.log(word);  
}
```

```
function execute(someFunction, value) {  
    someFuunction(value);  
}
```

```
execute(say, "hello");
```

```
$ node functions.js
```

```
functions.js:7  
    someFuunction(value);  
    ^
```

```
ReferenceError: someFuunction is not defined  
    at execute (functions.js:7:2)  
    at Object.<anonymous> (functions.js:10:1)  
    at Module._compile (module.js:449:26)  
    at Object.Module._extensions..js (module.js:467:10)  
    at Module.load (module.js:356:32)  
    at Function.Module._load (module.js:312:12)  
    at Module.runMain (module.js:492:10)  
    at process.startup.processNextTick.process._tickCallback (node.js:244:9)
```

```
$
```

Modules

- Libraries
- Include files
- Facilitates code organization
- Directory walk looking for 'node_modules/' to satisfy requires

Modules

```
fs = require('fs');
```

```
fs.rename(files.upload.path, "/tmp/test.png",  
  function(err) {  
    if (err) {  
      console.log('error');  
    }  
  });
```

Functions

```
function say(word) {  
    console.log(word);  
}
```

```
function execute(someFunction, value) {  
    someFunction(value);  
}
```

```
execute(say, "hello");
```

EventEmitters

In node.js an event can be described simply as a string with a corresponding callback. An event can be "emitted" (or in other words, the corresponding callback be called) multiple times or you can choose to only listen for the first time it is emitted.

EventEmitters

Many objects in Node emit events: a `net.Server` emits an event each time a peer connects to it, a `fs.readStream` emits an event when the file is opened. All objects which emit events are instances of `events.EventEmitter`. You can access this module by doing:
`require("events");`

Typically, event names are represented by a camel-cased string, however, there aren't any strict restrictions on that, as any string will be accepted.

Functions can then be attached to objects, to be executed when an event is emitted. These functions are called *listeners*.

EventEmitters

- on
- once
- removeListener
- once

Debugging

- `console.log`
- `debugger` (`node debug file.js`)

Tricks

- Rapid deployment
 - screen
 - node server.js
 - return to the session with 'screen -r'

Node in the EC2

- Excellent step-by-step tutorial on running node in the EC2
- <http://www.bennadel.com/blog/2321-How-I-Got-Node-js-Running-On-A-Linux-Micro-Instance-Using-Amazon-EC2.htm>

Live demo

- http server
- fixed protocol response
- directory list

Links

- nodejs.org
- npmjs.org
- github.com/joyent/node
- howtonode.org
- nodebeginner.org
- nodeguide.com
- expressjs.com
- github.com/dougtoppin/node