Node js

Intro to Javascript on the server

Agenda

- What is Node?
- Installing Node
- The global object
- loading modules
- EventEmitter
- Creating Node modules
- Working with the File System
- Making Requests
- Building web servers
- Posting form data
- NPM
- Using community modules
- connect
- Testing with Mocha and Chai

What is Node js

What

JavaScript without a Browser
Based on Chrome's V8 Engine
Officially Called “node”

How

Open source project
Sponsored by Joyent
Community written modules

What it can do

Act as a web server
Build command-line tools
Build socket applications
Anything you can do in JS
Ideal for simple “web stuff”

Node Source

github.com/joyent/node
Still on version 0

Node js Pros and Cons

Reasons to use

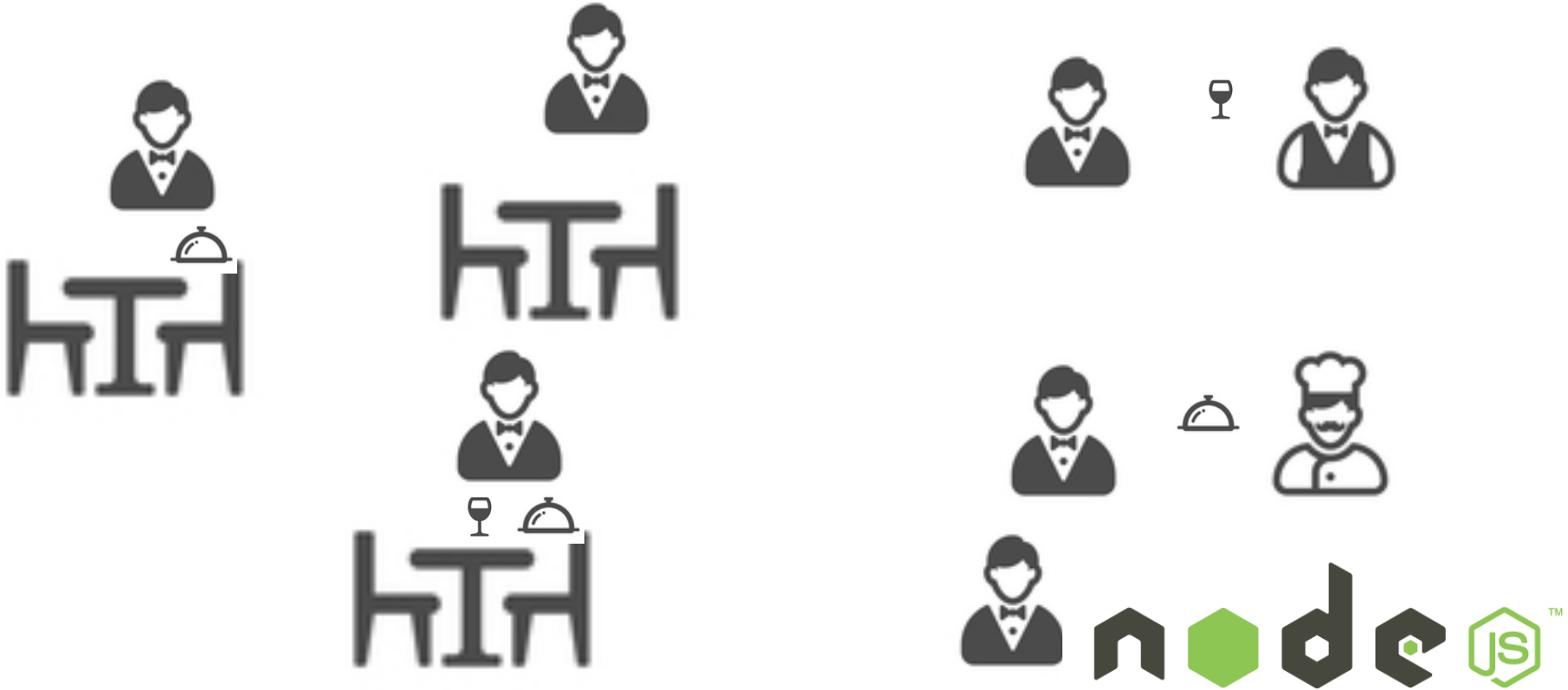
- Already JavaScript savvy
- IO heavy applications take advantage of non-blocking IO
- You love JavaScript
- Building a web service
- Wish you could code everything in JavaScript
- Reuse code between client and server

Reasons not to use

- Building command line tools
- Replacing shell scripts
- Intense, CPU bound applications
- You think JavaScript is horrible

Apache Steak House

Apache Steak House



Chez Node

How Node Works

- Single Thread
- Event Loop
- Respond to Events in Order
- Callback Functions
- Asynchronous
- Shared Global

Installing node

Platforms
Windows
OSX
Linux

Installing
run installer
(nodejs.org)

NVM
https://github.com/creationix/nvm

First node.js File

global

- console
- __filename, __dirname
- process
- module, exports

util

- util.puts()
- util.print()
- util.log()
- util.format()

Path Module

- path.normalize()
- path.resolve()
- path.dirname()
- path.basename()

INPUT

process.argv

An Array of arguments sent through the command line.

```
> node app.js Sending These Args
process.argv
["node",
"/root/myApp",
"Sending",
"These",
"Args"]
```

process.stdin

Standard input collected directly from the console.

```
> node myApp.js

Input: I typed this
Data Received: I typed this
Input: waiting for more input
```

Node Events

EventEmitter

Any time an object in node emits an event that object has inherited the EventEmitter.

```
var e = require('events');

var mYeller = new e.EventEmitter();

mYeller.on('yell', function(msg) {
      console.log( msg );
});

mYeller.emit('yell', 'Holla Back!');
```

Node Modules

app.js

```
var util = require('util');
var module1 = require('./module');

module1.publicNumber;     //  10
module1.publicMethod();   //   Hello World
module1.privateProperty;  // undefined
```

module.js

```
var privateProperty = null;
var privateMethod = function() {
        return "Hello World";
};

exports.publicNumber = 10;

exports.publicMethod = privateMethod;
```

Node JS File System

- Create Files
- Write to Files
- Read Files
- Move Files
- Remove Files
- Create Directories
- List files in Directories
- Move Directories
- Remove Directories

fs

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

fs.readFile( "FileName", function(err, data) {

        data ← The Buffer

});

fs.readFile( "File", "UTF-8", function(err, data) {

        data ← A String

});
```

http.requests

Request

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

var options = {
        hostname: 'nodejs.org',
        port: 80,
        path: '/',
        method: 'GET'
};

var callback = function(response) {

        response.on('event', fn);
};

var req = http.request(options, callback);
```

Response

response.statusCode
response.headers

Response Events

data	A new chunk of data is received
end	The response has ended
error	An error has occured

Creating Web Servers

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

http.createServer(function (req, res) {

   res.writeHead(200, {'Content-Type': 'text/plain'});

   res.end('Hello World\n');

}).listen(1337, '127.0.0.1');
```

req : Request

An object containing information about the request including request headers.

res : Response

An object that packages the response and lets us modify the response before sending it.

File Servers

```
fs.readFile('./file.html', 'UTF-8', function (err, data) {

    if (err) {
        throw err;
    }

    res.writeHead(200, {'Content-Type': 'text/html'});
    res.end(data);

});
```

Getting POST Data

Posted Form Data:

fav-color=%23000000&stoke-level=6

Posted File Data:

```
-   -----WebKitFormBoundaryGsTefag7g3LFBijB
    Content-Disposition: form-data; name="fav-color"

    #000000
    ------WebKitFormBoundaryGsTefag7g3LFBijB
    Content-Disposition: form-data; name="stoke-level"

    6
    ------WebKitFormBoundaryGsTefag7g3LFBijB
    Content-Disposition: form-data; name="upload-file"; filename="
working.js"
    Content-Type: application/javascript

    This is where the file body is placed...
```

```
if (req.method == 'POST') {

    var body = '';
    req.on('data', function (chunk) {
        body += chunk;
    });

    req.on('end', function () {
        // The body contains your data
    });

}
```

package.json

Package.json

- Manage App info
- Manage Dependences
- Incorporate Testing
- Incorporate Licensing

Creating Package.json

- just a json file
- npm init
- modify text directly
- npm install *modulename* --save

```
> npm init
name: (app)
version: (0.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (BSD)
```

npm installing a node module

$ npm install *module*

$ npm install *module* --save

$ npm install *module* --save-dev

$ npm install *module* --g

** Node modules are saved to the ./node_m*

$ npm install optimist

A node modules that takes terminal arguments and places them on a hash.

```
$ node arguments.js --adj nice --pastVerb ate --verb ran --noun car
```

```
var argv = require('optimist').argv;

argv.adj;              // 'nice'
argv.pastVerb;         // 'ate'
argv.verb;             // 'ran'
argv.noun;             // 'car'
```

$ npm install formidable

```
function (req, res) {
  var form = new formidable.IncomingForm();
  form.uploadDir = __dirname + "/Uploads/";
  form.parse(req, function (err, fields, files) {

        err;      // Any errors that may have occurred while parsing
        fields;  //  The form fields
        files;    //  Any uploaded files

  });
}
```

$ npm install connect@2.25

Connect Can help us easily server static files.

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

var app = connect().use(connect.static('public'));
http.createServer(app).listen(3000);
```

Unit Testing with Mocha

1) Install

$ npm install mocha

$ npm install should

1) Spec

place tests in directory ./test

name file *yourname*-spec.js

should js (github)

```
// Story
describe(story, callback);

// Test
it(test, callback);


//Assertion
obj.should.equal({});
```