

NODE.JS

LUNCH N' LEARN

Dave Stevens / dstevens@zappos.com / @shakefon

WHUT?

- Platform (not a language)
- Built upon V8 (Chrome's JS engine)
- JavaScript!
- Lightweight & Efficient
- Event-driven, non-blocking I/O model

OH, SERVER-SIDE JAVASCRIPT!

NO.

CLI

```
respond(process.argv);

function respond (args) {
  if (args.length < 3) {
    console.log('Come on, give me something to work with');
    process.exit(1);
    return;
  }

  if (args[2] == 'help') {
    console.log('You need help!');
    process.exit(0);
    return;
  }
}
```

IRC

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

var irc = ircb({
  host: 'irc.freenode.org',
  secure: true,
  nick: 'nodeIRCBot',
  username: 'nodeIRCBot',
  realName: 'nodeIRCBot'
}, function () {
  console.log('Connected');
  console.log('MOTD:\n');
  console.log(irc.motd);
});
```

AND YES, A WEB SERVER...

NON-BLOCKING?

- Asynchronous code execution
- Embrace callbacks!
- Don't rely on order of execution

TRADITIONAL BLOCKING I/O

PHP

```
<?php

$results = databaseQuery("SELECT * FROM massive_table");
var_dump($results);

// Once all results are returned, databaseQuery finishes
// executing and returns the results, then we see them
// printed out.

// But what if that query takes a long time to run?
```

NON-BLOCKING I/O

NODE.JS

```
var results;

databaseQuery('SELECT * FROM massive_table', function (returnedRows) {
  results = returnedRows;
});

console.log(results);

// console.log will be executed immediately after the databaseQuery
// func is invoked, so that will be an undefined value.

// But, execution is not blocked - so if the query takes a long time,
// the rest of the program can continue. Code accordingly!
```

MAKE USE OF YOUR CALLBACK FUNCTION

```
var results;  
  
databaseQuery('SELECT * FROM massive_table', function (returnedRows) {  
    results = returnedRows;  
    console.log(results);  
});  
  
// now we have results logged out when they're ready, but anything  
// else we need to do in parallel isn't blocked waiting for the results
```

NPM

NODE PACKAGE MANAGER

A repository of community-contributed modules (you can contribute!) to perform a variety of tasks, so that the core of node remains uncluttered and minimal.

PACKAGE.JSON

Tells npm about your project, including any module dependencies that you have. Needed for publishing to npm's repository but also very useful for deploying your project...

PACKAGE.JSON

```
{
  "name": "Example Module",
  "version": "0.0.1",
  "author": "Dave Stevens <dstevens@zappos.com>",
  "description": "An excuse to show off a package.json file",
  ...
  "dependencies": {
    "mysql": "2.0.x",
    "winston": "*"
  }
  ...
}
```

Interactive guide at <http://package.json.jit.su>

USING NPM

Install someone else's module:

```
npm install mysql
```

USING NPM

Update installed module:

```
npm update mysql
```


USING NPM

Install your project:

```
cd /path/to/your/project  
npm install
```

WEB SERVER

Mindshift if you're from a PHP / Java / etc background -
you're now handling the application AND the server in one
place

Listens for connections on a port, so file path is largely
irrelevant

WEB SERVER

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

http.createServer(function (req, res) {
  res.writeHead(200, {"Content-Type": "text/plain"});
  res.write("Hello, world!");
  res.end();
}).listen(3000);
// Chaining!
console.log("Server running - http://localhost:3000");
```

EXPRESS

- Web Application Framework
- Connect middleware
- Actively developed, mature (for anything node...) and in use today here

EXPRESS

INSTALLING

```
npm install express
```

EXPRESS

SIMPLE WEB APP

```
var express = require('express');
var app = express();

app.get('/', function (req, res) {
  res.send("Hello, world! Zoooooom!");
});

app.listen(3000);
console.log("Express app running - http://localhost:3000");
```

EXPRESS

SIMPLE WEB APP

```
var express = require('express');
var app = express();

app.get('/', function (req, res) {
  res.send("Hello, world! Zoooooom!");
});

app.get('/more-original', function (req, res) {
  res.send("Because Hello World is just too cliched");
});

app.listen(3000);
console.log("Express app running - http://localhost:3000");
```

BUILDING PROJECTS

Grunt.js is written in node, can also be used to build your projects, do minification and concatenation, etc

Assemble (npm install assemble) lets you do static file generation like Jekyll, but in node

COOL PROJECTS IN NODE

- Nodecopter
- Nodebots
- AirBnB
- LinkedIn Mobile

STYLE GUIDE

[HTTPS://GITHUB.COM/FELIXGE/NODE-STYLE-GUIDE](https://github.com/felixge/node-style-guide)

Get in to some good habits

NEXT STEPS

- Tech U class soon
- Embrace IRC! #node.js
- Resources to be sent out today
- Heroku, AWS Free Tier
- All you need is your laptop!

OBLIGATORY QUESTIONS SLIDE

Love talking about node, come talk anytime

- dstevens@zappos.com
- [@shakefon](#)