



Peter Elger  
@pelger

# Today

- A whirlwind Tour of node
  - node.js
  - npm eco-system
  - MEAN stack
- Some code!
  - MEAN stack MLS example





# Node.js



- If you want to play along later then:
  - Go to <http://nodejs.org>
  - Download and install the binary for your platform
  - Do it now!! (if you haven't already 😊 )



# Node.js

- Node.js is a highly efficient and scalable non-blocking I/O platform that was build on top of Google Chrome V8 engine and ECMAScript
- i.e. server side Javascript done right

# Node.js

- Runs high-performance server-side JavaScript
- Uses the Google Chrome V8 engine
  - just-in-time compilation to machine code
  - generation garbage collection (like the Java JVM)
  - creates virtual “classes” to optimize property lookups
- Provides a minimal system level API for networking, file system, event handling, streaming data, and HTTP/S.
- Has a well-designed module system for third party code - very effective and simple to use
- Your code runs in a single non-blocking JavaScript thread
- That’s OK, most of the time you’re waiting for the database or network anyway!



# Blocking Code

- Traditional code waits for input before proceeding

```
Statement stmt = conn.createStatement();  
ResultSet rs = stmt.executeQuery("SELECT * FROM Customers");
```

- The Java thread above "blocks" on executeQuery
- Execution does proceed until database returns a result
- The thread consumes large amounts of resources:
  - memory for the stack
  - processing time for context switching

# NonBlocking Code

- Callback-based code waits for events

```
collection.findOne( query, function( err, result ) { ...
```

global object obj

- Executing the callback is cheap
  - just a JavaScript function call with a given context
- There is no thread management
  - just execute callbacks when there are events
- Callback functions are easier to use when the language supports anonymous functions



# Major differences to browser

- Global scope – removed
  - There is a global object – use is highly discouraged
- No DOM, window or document object
  - Although this environment can be created
- File / module inclusion
  - require – to load a file / module
  - exports – to export an object for use with require
- Additional core APIs e.g FileSystem, Streams etc...

# Hello World

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

http.createServer( function (request, response) {
  response.writeHead(200, {'Content-Type': 'text/plain'});
  response.end('Hello World\n');
}).listen(8000);

console.log('Server running at http://localhost:8000/');
```



# Non-blocking, evented I/O

- Node is single threaded and transparently runs an event loop on process start
- scaling on multi-core is achieved through balancing across processes
- Primary async mechanisms are the callback and EventEmitter



# Callback Pattern

```
var f = function(params, callback) {  
  // do work..  
  if (/*an error occurs */) {  
    return callback(new Error("An error has occurred"));  
  }  
  callback(null, results);  
}
```

```
f(params, function(err, results) {  
  if (err) ...  
});
```



# EventEmitter Pattern

```
var events = require('events');  
var emitter = new events.EventEmitter();  
  
var listener = function() {  
  console.log('fired');  
};  
emitter.on('myevent', listener);  
emitter.emit('myevent');
```



# npm

- THE module management and distribution system for node
  - The apt-get / CPAN / PEAR for node
- Provides a command line interface to search and install modules from the public npm registry
  - <http://npmjs.org>.
- A node module is a set of files described by a package.json



# package.json

```
{
  'name': 'mlshack',
  'description': 'my awesome project',
  'license': 'MIT',
  'author': 'Peter Elger (http://www.nearform.com/)',
  'main': 'lib/main',
  'version': '0.0.1',
  'dependencies': {
    'express': '~3.4.3',
  },
  'devDependencies': {
    'grunt': '~0.4.1',
  },
  'engines': {
    'node': '>=0.10.0'
  },
}
```



# Basic Commands

- Create a new package.json

```
$npm init
```



- Install modules

```
$npm install <module-name> --save
```

- Install deps

```
$npm install
```

- Publish your work!!

```
$npm publish
```





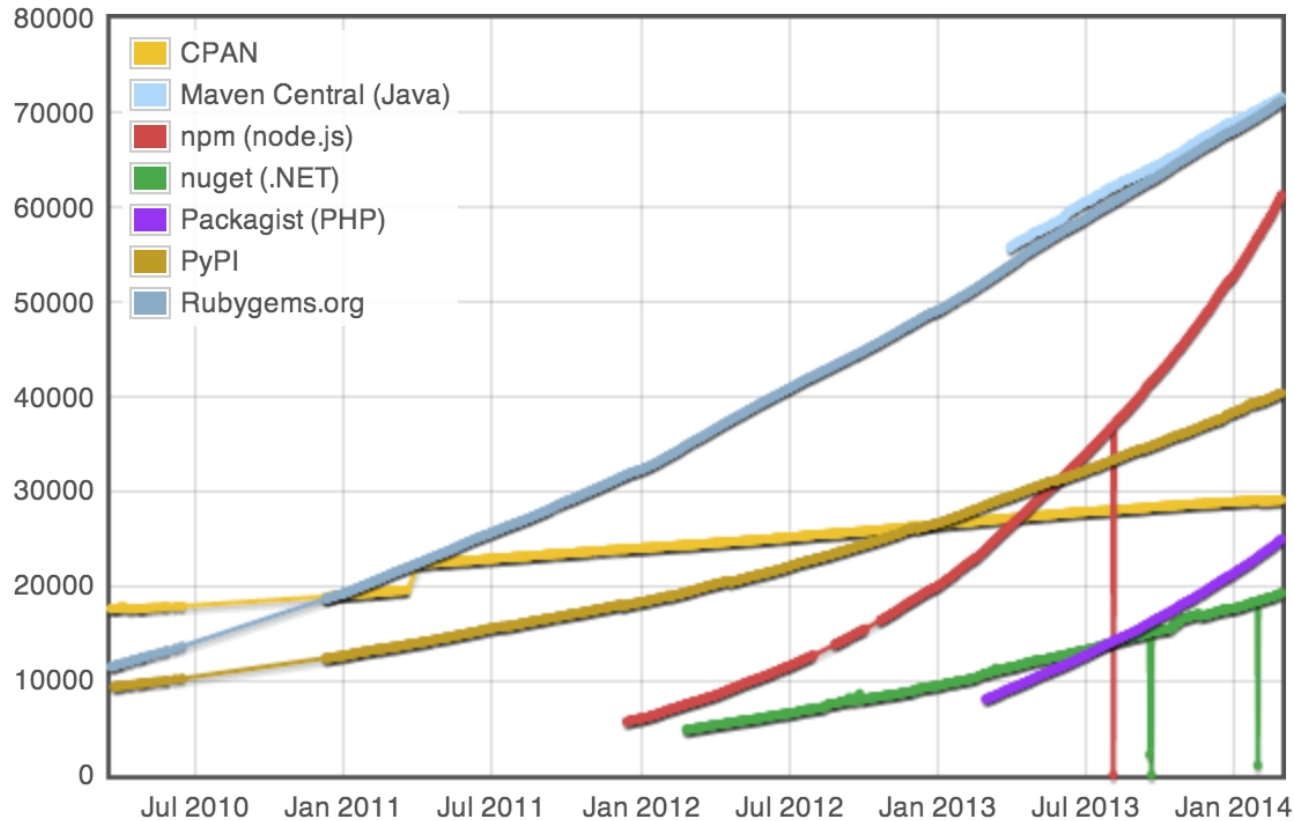
# There's a module for that...

- For most tasks that you want to achieve there is usually a node module available
- Google is one way...
- From nearForm nodezoo – page rank for node modules:
  - <http://nodezoo.com/>



# Growth

## Module Counts



### Include

- Clojars (Clojure)
- CPAN
- CPAN (search)
- CRAN (R)
- Hackage (Haskell)
- Maven Central (Java)
- MELPA (Emacs)
- npm (node.js)
- nuget (.NET)
- Packagist (PHP)
- Pear (PHP)
- PyPI
- Rubygems.org

# MEAN Stack

- A common use case for node is the development of 'single page applications' backed by a NoSQL data store
- MEAN === Mongo, Express, Angular, Node



# MEAN

- **MongoDB** (from "humongous") is an open-source NoSQL document database
- **Express** is a minimal and flexible node.js web application framework.
- **Angular** client side application framework that lets you extend HTML vocabulary
- **Node.js**





My Feelings About AngularJS Over Time

<http://www.bennadel.com/>

# Now some fun!

THE WEB'S SCAFFOLDING TOOL FOR MODERN WEBAPPS



# YEOMAN

# Workflow

- Yo – scaffold out application
- Grunt – build, preview and test
- Bower – front end dependency management
- Npm – back end dependency management



# Setup

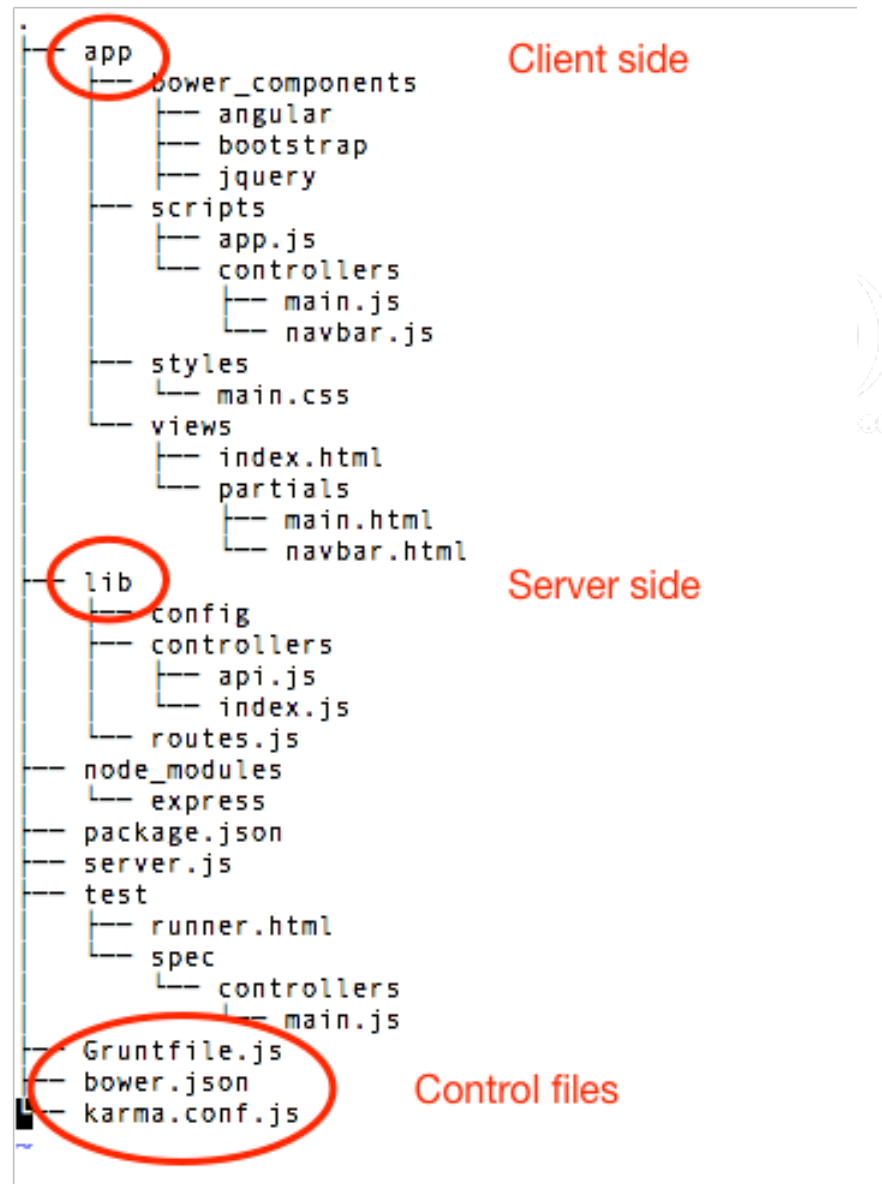
```
$npm install -g yo
```

```
$npm install -g grunt-cli
```

```
$npm install -g bower
```

```
$npm install -g generator-angular-  
fullstack
```





@nearform  
cian.omaidin@nearform.com  
richard.rodger@nearform.com  
anton.whalley@nearform.com



[www.nearform.com](http://www.nearform.com)