

#PTW16 Intro to Node.JS

Ross Kukulinski



NODESOURCE®

May 4, 2016

@RossKukulinski

Product Manager – NodeSource

Node Foundation Evangelist

Container Advocate & Consultant

Introduction to CoreOS - O'Reilly Media



Overview

phillydev.org

#ptw-nodejs-workshop



Why are you here?

Workshop Goals

- Learn what Node.JS *is* and *is not*
- Understand non-blocking I/O and event-driven programming
- Hands on with Node.JS!
- Prepare you for future Node.JS learning

Schedule

- 9:00 - 9:05 — Intro & Overview
- 9:05 - 9:45 — What *is* Node.JS
- 9:45 - 10:00 — Hands-on with NodeJS
- 10:00 - 10:30 — NodeSchool Pair-Programming
- 10:30 - 10:45 — Coffee Refill
- 10:45 - 11:00 — Questions / Review
- 11:00 - 11:45 — Continued Pair-Programming
- 11:45 - 12:00 — Wrap up & Summary

What is Node.JS



A screenshot of a web browser window showing the Node.js homepage at <https://nodejs.org/en/>. The page features the Node.js logo at the top left, followed by a navigation bar with links for HOME, ABOUT, DOWNLOADS, DOCS, FOUNDATION, GET INVOLVED, SECURITY, and NEWS. Below the navigation bar, a main text block describes Node.js as a JavaScript runtime built on Chrome's V8 engine, using an event-driven, non-blocking I/O model. A green callout box highlights an "Important security notification regarding npm". Two large download buttons are shown: one for "v4.4.3 LTS" (Recommended For Most Users) and another for "v6.0.0 Current" (Latest Features). At the bottom, links for "Other Downloads", "Changelog", and "API Docs" are provided for both versions.

Node.js

<https://nodejs.org/en/>

node

HOME | ABOUT | DOWNLOADS | DOCS | FOUNDATION | GET INVOLVED | SECURITY | NEWS

Node.js® is a JavaScript runtime built on [Chrome's V8 JavaScript engine](#). Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, [npm](#), is the largest ecosystem of open source libraries in the world.

Important [security notification](#) regarding npm

Download for OS X (x64)

v4.4.3 LTS
Recommended For Most Users

v6.0.0 Current
Latest Features

Other Downloads | Changelog | API Docs Other Downloads | Changelog | API Docs

Or have a look at the [LTS schedule](#).

What is Node.JS

Node.JS is a JavaScript runtime
built on Chrome's JavaScript
engine* for building fast, scalable
network applications.

Node.JS is NOT

... a language

... a web framework



What is Node.JS

Node by the Numbers

- Node.JS is the fastest growing development platform
- 3.5 million users and annual growth rate of 100%
- npm is the largest ecosystem of open source libraries in the world



What is Node.js



Enterprises have
already adopted
Node.js

Why does Node.JS exist?



Why does Node.JS Exist?

Latency: Memory vs I/O

L1 cache	0.5ns
L2 cache	7ns
Main memory	100ns
Send 1K bytes over 1 Gbps network	10,000ns
Round trip within same datacenter	500,000ns
Disk seek	10,000,000ns
Packet CA->Netherlands->CA	150,000,000ns



Why does Node.JS Exist?

I/O is Expensive



Blocking vs Non-Blocking

Blocking I/O

```
console.log('Fetching article...');

var result = query("SELECT * FROM articles WHERE id = 1");

console.log('Here is the result:', result);
```



```
1 // Java: Spot the I/O
2
3 System.out.println("Reading file...");
4 BufferedReader br = new BufferedReader(new
5           FileReader("in.txt"));
6
7 try {
8     StringBuilder sb = new StringBuilder();
9     String line;
10
11    while ((line = br.readLine()) != null)
12        sb.append(line + "\n");
13    System.out.print(sb.toString());
14 } finally {
15     br.close();
16 }
17
18 System.out.println("Finished reading file!");
```



Blocking vs Non-Blocking

Some solutions

- One process per connection (Apache Web Server 1)
- One thread per connection (Apache Web Server 2)
- Event-Driven (nginx, node.js)



Event-Driven, Non-Blocking Programming

```
function handleResult(result) {  
    console.log('Here is the result:', result);  
}  
  
select('SELECT * FROM articles WHERE id = 1', handleResult);  
  
console.log('Fetching article...');
```



Why JavaScript?

JavaScript well-suited to event-driven model



Functions are First-Class Objects

Treat functions like any other object

- Pass them as arguments
- Return them from functions
- Store them as variables for later use



Functions are Closures

```
function calculateSum (list) {  
    var sum = 0;  
  
    function addItem(item) {  
        sum += item;  
    }  
  
    list.forEach(addItem);  
  
    return function() { // return a function that can print the result  
        console.log('sum: ' + sum);  
    }  
}  
var printSum = calculateSum([ 2, 5, 10, 42, 67, 78, 89, 120 ]);  
  
// printSum is a function returned by calculateSum but it still  
// has access to the scope within its parent function  
printSum();
```



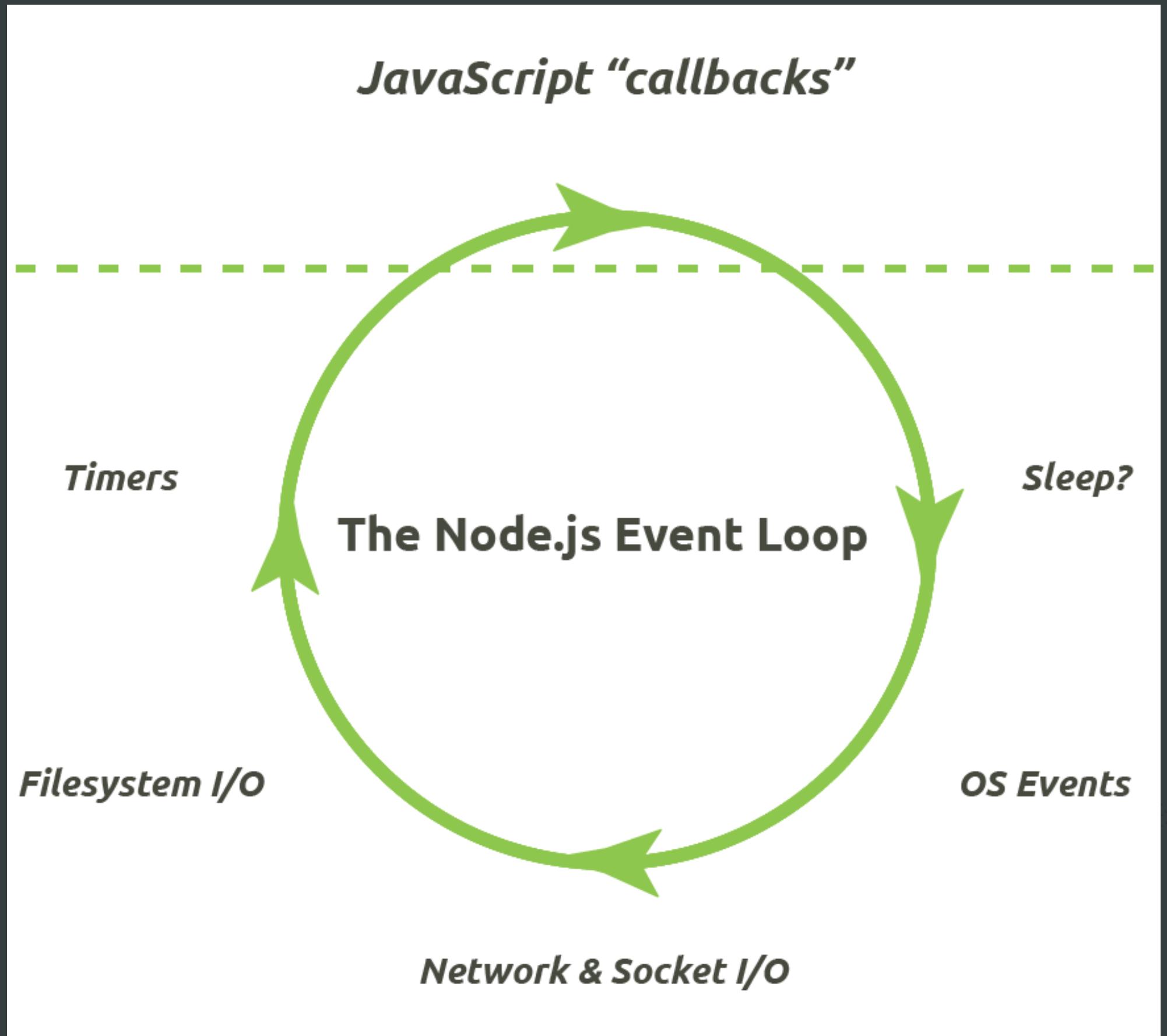
Node.JS Non-Blocking I/O

- Kernel-level non-blocking socket I/O using epoll() or select() system calls (depending on the operating system)
- File system I/O delegated to worker threads
- Glued together with an “event loop”



Node.JS Event Loop

- Perform work triggered by JavaScript code
- Perform work triggered by external events
- Pass events & data to JavaScript via callback functions
- Exit when no more pending work or events



When does it stop?

```
// 1.  
fs.readFile('bigdata.txt', 'utf8', function (err, data) {  
  console.log(data.split('\n').length, 'lines of data');  
});
```

When does it stop?

```
// 2.  
setTimeout(function () {  
  console.log('Waited 10s');  
}, 10000);
```

When does it stop?

```
// 3.  
var i = 0;  
var timer = setInterval(function () {  
    console.log('tick', i);  
    if (++i === 60)  
        clearInterval(timer);  
}, 1000);
```


Sync vs Async

Synchronous

```
var bigdata = fs.readFileSync('bigdata', 'utf8');
```

Asynchronous

```
fs.readFile('bigdata', 'utf8', function (err, bigdata) {  
  // ...  
});
```



Rules of Thumb

- Choose asynchronous I/O over synchronous I/O
- Opt for parallel I/O wherever possible
- Don't hog the JavaScript thread with long-running calculations or iterations

Node.JS is Minimal

- Complexity is in user-land
- Instead of a big framework, Node.JS provides minimal viable library for doing I/O
- All the rest is built on top in user-land
- Pick and choose from HUGE library of third-party modules

npm - Node Package Manager

<https://www.npmjs.com/>

250,000+ packages of reusable code



Node.JS Strengths

Node is great for:

- REST APIs
- Mobile Backend
- Real-time applications
- Streaming services
- Prototyping
- Front-end development



Node is NOT good for:

- CPU-bound workloads
- Serving static content

Hands-on with Node.JS



Installing Node.JS

Windows

<https://nodejs.org/en/download/>

OSX/Linux

<https://github.com/creationix/nvm>

Installing Node.JS

What do you get?

- node – The Node.JS runtime binary
- npm – CLI tool allows you to use the extensive module ecosystem



Example Code

github.com/rosskukulinski/ptw16_node



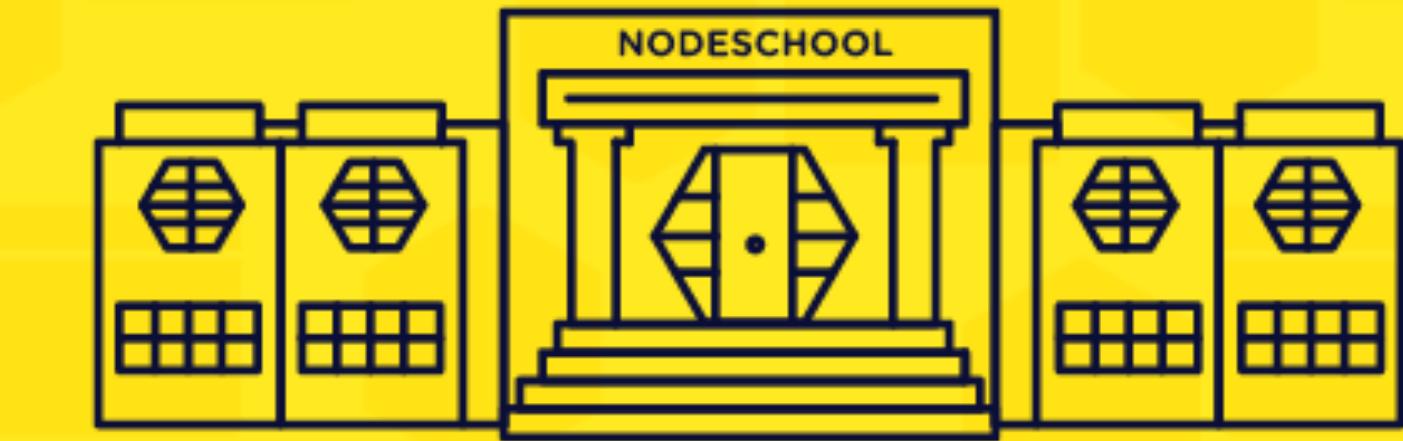
Live demo time!

NodeSchool Pair-Programming



Workshoppers

Open source lesson modules to teach JavaScript, Node.js, npm, and many more topics. All lessons are self-guided and most work offline.



NODESCHOOL

Open source workshops that teach web software skills.

Do them on your own or at a workshop nearby.

Find a Partner!

Getting started

1. npm install -g learnyounode
2. Make a folder to store your code
3. learnyounode

Let's work through the
first exercise together

Break time! Back at 10:45



Questions / Review



Back to it!



Wrap Up



Resources for future learning

- <http://nodeschool.io>
- <https://nodejs.org/api/>
- <https://github.com/nodejs/help>
- #node.js on irc.freenode.net
- <http://nodeweekly.com/>
- <http://planetnodejs.com/>
- <http://nodeup.com/>
- <http://stackoverflow.com/questions/tagged/node.js>
- <http://www.meetup.com/nodejs-philly/>

Package recommendations

- Logging: Bunyan, winston
- Webserver: Express, hapi, kraken
- REST APIs: restify, hapi
- Testing: mocha, tap, jasmine
- Frontend: browserify, gulp, grunt, webpack
- Real-time: socket.io, ws

Also experiment with functional programming (Node.JS Streams)

Wrap Up

Want to get involved?

- <https://nodejs.org/en/get-involved/>
- Help with documentation
- Contribute to Node Core
- Contribute to npm modules
- Run your own NodeSchool



The Enterprise Node Company™

<https://nodesource.com>



N|Solid: Enterprise Grade Node

- Production Monitoring
- Performance Analysis
- Platform Security
- Free for Development



N | Solid™

Thank you.

Ross Kukulinski

ross@nodesource.com

@RossKukulinski



NODESOURCE®