



Intro to NodeJS HackBU

Clay Himmelberger
March 27, 2018



Outline

What is NodeJS

Why NodeJS

Coding Examples and Demo



What is NodeJS

Straight from nodejs.org

“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.”

Let's break that down...



What is NodeJS

“JavaScript runtime built on Chrome's V8 JavaScript engine.”

This is JavaScript on the server.

Still a scripting language, so not compiled.

Quick startup, lower resources.

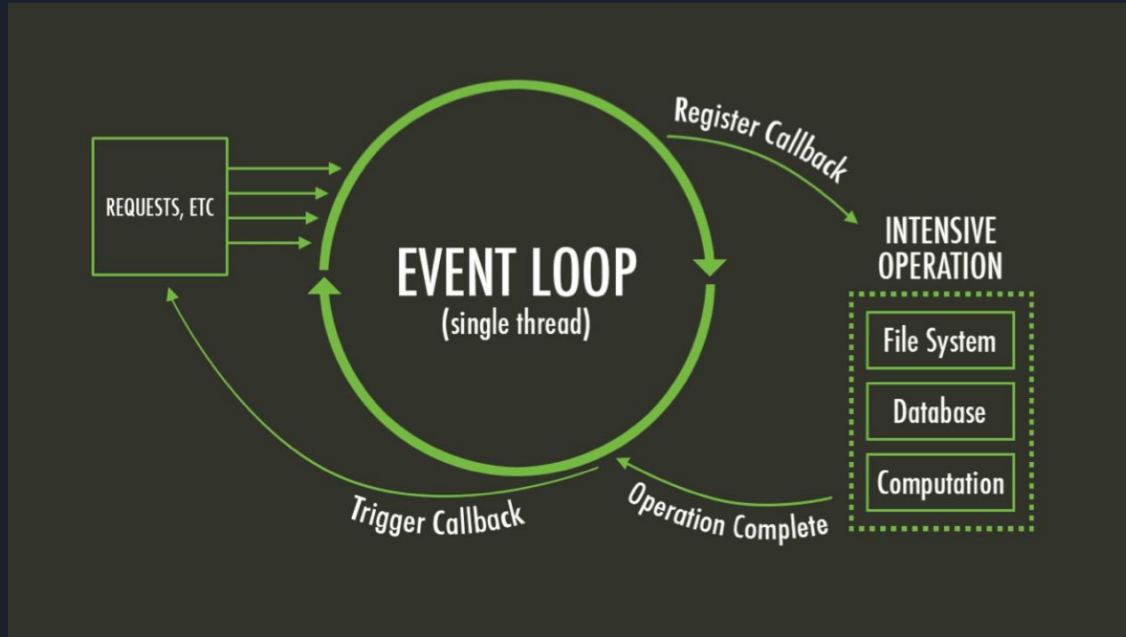
Same language used in browsers everywhere - developers are familiar.

What is NodeJS

“Node.js uses an event-driven...”

NodeJS is **single-threaded**, run by an Event Loop.

From MDN: “Each message is processed completely before any other message is processed. This offers some nice properties when reasoning about your program, including the fact that whenever a function runs, it cannot be pre-empted and will run entirely before any other code runs (and can modify data the function manipulates).”





What is NodeJS

“Node.js uses an event-driven, non-blocking I/O...”

Also from MDN: “A very interesting property of the event loop model is that JavaScript, unlike a lot of other languages, never blocks.”

If you're more interested, look at [libuv](#)

For any I/O in the program (file system, socket connections, services like a database, user input) , other processing is not impacted. Highly efficient for 1,000's of web connections or other heavy I/O use, with much lower overhead and server requirements.



What is NodeJS

“Node.js' package ecosystem, [npm](#), is the largest ecosystem of open source libraries in the world.”

A program in NodeJS defines its dependencies in its `package.json` file. A description of the program, common scripts for the program, build dependencies and development dependencies.

(kind of like a Java Maven file, Ruby Gems, PyPI, etc)

Makes for easy building and version control of needed packages.

As of 3-20-2018: “470,000 free code packages in the npm Registry”



Why NodeJS

Npm ecosystem - embraced open-source

Popular and adopted by big companies

- [StackOverflow 2018 Developer Survey](#) - 70% JS, ~50% NodeJS, 50k+ responses.
- [Hired.com puts Javascript](#) at top of skills to know
- Used by big names in tech - Netflix, LinkedIn, Walmart, Trello, Paypal

Less mental “context switching” between server and web front-end.

JSON messages, which are pretty expressive and efficient, easily communicate from front-end to back-end.

Great for “message passing” systems, even in real-time.



Why Not NodeJS

CPU intensive applications - it's single-threaded

It's JavaScript, and not compiled. Type-checking, typo-checking, code-lookup are basically out the window, and it crashes at run-time. (TypeScript and JS Doc are bridging this gap, but it's no Java/C#... write those unit tests!)

Exceptions crash your process . . . and it's single-threaded so production code needs babysitting.

Lots of open source, lots of other packages. Huge build chains. You might not know what 80% of your true code base is doing! Is it maintained? Is it correct? Is it dependant on other things that are maintained and correct? Look up “node left pad” for an early-2016 example of amazing automation in chaos.



Enough talk . . .

Code Example and Demo

<https://github.com/chimmelb/intro-nodejs-hackbu>

Links that are helpful:

[NodeJS Docs](#)

[MDN Javascript](#)

[Stack Overflow Questions](#)