

How do I get started with Node.js

Are there any good resources to get started with Node.JS? Any good tutorials, blogs or books?

Of course, I have visited its official website <http://nodejs.org/>, but I didn't think the documentation they have is a good starting point.

javascript node.js

edited Jan 13 '13 at 16:05

community wiki
8 revs, 5 users 48%
Joneph O.

locked by Shog9 ♦ Jul 19 '13 at 22:30

This question's answers are a collaborative effort: if you see something that can be improved, just edit the answer to improve it! *No additional answers can be added here*

8 If and of you are wondering on how to build a website using node.js and you're coming from a php'ish background, I've asked how to do *that* here stackoverflow.com/questions/11311672/... I feel that's something a lot of people miss. – Benjamin Gruenbaum Mar 8 '13 at 7:53

3 Answers

Tutorials

- [NodeSchool.io interactive lessons](#)
- [The Art of Node \(an introduction to Node\)](#)
- [Hello World](#)
- [Hello World Web Server \(paid\)](#)
- [Node.js guide](#)
- [Build a blog with Node.js, express and MongoDB](#)
- [Node.Js Tutorials At Project 70](#)
- [Node.js for Beginners](#)
- [Learn Node.js Completely and with Confidence](#)
- [Absolute Beginners Guide To Node.js](#)
- [Node JS Processing Model – Single Threaded Model with Event Loop Architecture](#)
- [RisingStack's Node Hero Series](#)

Developer Sites

- [Joyent's developer site for node](#)

Videos

- [Node tuts](#)
- [Introduction to Node.js with Ryan Dahl](#)
- [Node.js: Asynchronous Purity Leads to Faster Development](#)
- [Parallel Programming with Node.js](#)
- [Server-side JavaScript with Node, Connect & Express](#)
- [Node.js First Look](#)
- [Node.js with MongoDB](#)
- [Ryan Dahl's Google Tech Talk](#)
- [Real Time Web with Node.js](#)
- [Node.js Tutorials for Beginners](#)
- [Pluralsight courses \(paid\)](#)

Screencasts

- [Learn All The Nodes](#)
- [NodeTuts](#)
- [NodeCasts](#)
- [Create server-side MVC apps with Node.js and Express](#)

Books

- [The Node Beginner Book](#)

- [Mastering Node.js](#)
- [Up and Running with Node.js](#)
- [Node.js in Action](#)
- [Smashing Node.js: JavaScript Everywhere](#)
- [Node.js & Co. \(in German\)](#)
- [Sam's Teach Yourself Node.js in 24 Hours](#)
- [Most detailed list of free JavaScript Books](#)
- [Mixu's Node Book](#)
- [Node.js the Right Way: Practical, Server-Side JavaScript That Scale](#)
- [Beginning Web Development with Node.js](#)
- [Node Web Development](#)
- [NodeJS for Righteous Universal Domination!](#)

Courses

- [Real Time Web with Node.js](#)
- [Essential Node.js from DevelopMentor](#)

Blogs

- [The Node.js blog](#)
- [How To Node](#)
- [DailyJS](#)
- [Nodejitsu blog](#)
- [Ryan Wilcox's Whitepaper](#)
- [devthought](#)

Podcasts

- [NodeUp](#)

JavaScript resources

- [Crockford's videos](#) (must see!)
- [Essential JavaScript Design Patterns For Beginners](#)
- [JavaScript garden](#)
- [JavaScript Patterns](#) book
- [JavaScript: The Good Parts](#) book

Node.js Modules

- [Search for registered Node.js modules](#)
- [A curated list of awesome Node.js libraries](#)
- [Wiki List on GitHub/Joyent/Node.js](#) (start here last!)

Other

- [JSApp.US](#) - like jsfiddle, but for Node.js
- [Node with VJET JS](#) (for Eclipse IDE)
- Production sites with published source:
 - [Node Knockout Hackathon \(source\)](#)
- [Useful Node.js Tools, Tutorials and Resources](#)
- [Runnable.com](#) - like jsfiddle, but for server side as well
- [Getting Started with Node.js on Heroku](#)
- [Getting Started with Node.js on Open-Shift](#)
- [Authentication using Passport](#)

edited Jul 27 at 21:14

community wiki
60 revs, 47 users 19%
[Kev](#)

17 Are the books ordered by your value in them or randomly? Looking to purchase a good reference manual on nodeJS's basics. – [David](#) May 17 '11 at 15:11

5 @David: More or less randomly. First two are freely available and the last one is in preview mode. I would recommend to start with The Node Beginner. – [yojimbo87](#) May 17 '11 at 15:18

99 My book (Node up and running) is available for free here: ofps.oreilly.com/titles/9781449398583 forever. It's also now an ebook and print. – [sh1mmer](#) May 20 '12 at 19:36

39 Though I'm sure it took a lot of effort to compile this list, it would actually have been a lot more helpful to get no more than 3 of the best places (in your opinion) – [Etai Raz](#) Nov 8 '12 at 6:58

5 @sh1mmer: Dead link. This one works (same book): chimera.labs.oreilly.com/books/1234000001808/index.html – [Nepoxx](#) Dec 10 '14 at 15:38

First, learn the core concepts of Node.js:

- You'll want to understand the asynchronous coding style that Node encourages.
- Async != concurrent. Understand Node's event loop!
- Node uses CommonJS-style `require()` for code loading; it's probably a bit different from what you're used to.
- Familiarize yourself with Node's standard library.

Then, you're going to want to see what the community has to offer:

The gold standard for Node.js package management is [NPM](#).

- It is a command line tool for managing your project's dependencies.
- Make sure you understand how Node and NPM interact with your project via the `node_modules` folder and `package.json`.
- NPM is also a registry of pretty much every Node.js package out there

Finally, you're going to want to know what some of the more popular packages are for various tasks:

Useful Tools for Every Project:

- [Underscore](#) contains just about every core utility method you want.
- [Lo-Dash](#) is a clone of Underscore that aims to be faster, more customizable, and has quite a few functions that Underscore doesn't have. Certain versions of it can be used as drop-in replacements of Underscore.
- [TypeScript](#) makes JavaScript considerably more bearable, while also keeping you out of trouble!
- [JSHint](#) is a code-checking tool that'll save you loads of time finding stupid errors. Find a plugin for your text editor that will automatically run it on your code.

Unit Testing:

- [Mocha](#) is a popular test framework.
- [Vows](#) is a fantastic take on asynchronous testing, albeit somewhat stale.
- [Expresso](#) is a more traditional unit testing framework.
- [node-unit](#) is another relatively traditional unit testing framework.
- [AVA](#) is a new test runner with Babel builtin and runs tests concurrently.

Web Frameworks:

- [Express.js](#) is by far the most popular framework.
- [Koa](#) is a new web framework designed by the team behind Express.js, which aims to be a smaller, more expressive, and more robust foundation for web applications and APIs.
- [sails.js](#) the most popular MVC framework for node, and is based on express. It is designed to emulate the familiar MVC pattern of frameworks like Ruby on Rails, but with support for the requirements of modern apps: data-driven APIs with a scalable, service-oriented architecture.
- [Meteor](#) bundles together jQuery, Handlebars, Node.js, [WebSocket](#), [MongoDB](#), and DDP and promotes convention over configuration without being a [Ruby on Rails](#) clone.
- [Tower](#) (*deprecated*) is an abstraction of top of Express.js that aims to be a Ruby on Rails clone.
- [Geddy](#) is another take on web frameworks.
- [RailwayJS](#) is a Ruby on Rails inspired MVC web framework.
- [Sleek.js](#) is a simple web framework, built upon Express.js.
- [Hapi](#) is a configuration-centric framework with built-in support for input validation, caching, authentication, etc.
- [Trails](#) is a modern web application framework. It builds on the pedigree of [Rails](#) and [Grails](#) to accelerate development by adhering to a straightforward, convention-based, API-driven design philosophy.
- [Danf](#) is a full-stack OOP framework providing many features in order to produce a scalable, maintainable, testable and performant applications and allowing to code the same way on both the server (node.js) and client (browser) sides.
- [Derbyjs](#) is a reactive full-stack javascript framework. They are using patterns like reactive programming and isomorphic JS for a long time.
- [Loopback.io](#) is a powerful Node.js framework for creating APIs and easily connecting to backend data sources. It has an Angular.js SDK and provides SDKs for iOS and Android.

Web Framework Tools:

- [Jade](#) is the HAML/Slim of the Node.js world
- [EJS](#) is a more traditional templating language.

- Don't forget about [Underscore's template method](#)!

Networking:

- [Connect](#) is the Rack or WSGI of the Node.js world.
- [Request](#) is a very popular HTTP request library.
- [socket.io](#) is handy for building WebSocket servers.

Command Line Interaction:

- [minimist](#) just command line argument parsing.
- [Yargs](#) is a powerful library for parsing command-line arguments.
- [Commander.js](#) is a complete solution for building single-use command-line applications.
- [Vorp.js](#) is a framework for building mature, immersive command-line applications.
- [Chalk](#) makes your CLI output pretty.

Work with streams:

- [mississippi](#) everything you miss about streams.
- <https://github.com/calvinmetcalf/streams-a-love-story>
- <http://maxogden.com/node-streams.html>
- <https://github.com/substack/stream-handbook>

edited Jul 27 at 21:10

community wiki
28 revs, 23 users 62%
Nevir

6 under frameworks you should add [railwayjs.com](#) – [sagivo](#) Jun 22 '12 at 14:55

14 @Farm question was about node.js. Angular is a browser framework and is irrelevant here. – [Alexander Ulitin](#) Nov 28 '13 at 6:32

Promise: To write maintainable and readable async code I would recommend [github.com/kriskowal/q](#) – [Farm](#) Dec 25 '13 at 6:36

Backbone.js, MEAN, Require.js and lots more to come !!! voila!!! – [Dileephehl](#) Jan 9 '14 at 10:47

NodeJS is the future of javascript – [Tarek Kalaji](#) Jan 14 '15 at 16:40

Use the source, Luke.

No but seriously I found that building Node from source, running the tests, and looking at the benchmarks did get me on the right track. From there, the .js files in the lib directory are a good place to look, especially the file http.js.

Update: I wrote this answer over a year ago, and since that time there has explosion in the number of great resources available for people learning node. Though I still believe diving into the source is worthwhile, I think that there are now better ways to get started. I would suggest some of the books on node that are starting to come out.

edited Apr 29 '11 at 15:43

community wiki
[Zach B](#)

6 +1 for the opening line :3 but thats how I leant to use Kohana. So it is a pretty good method as long as you can understand the source. – [Olical](#) Mar 15 '11 at 15:19

66 +1 for updating your answer a year after you wrote it. – [Caleb](#) May 3 '11 at 14:26

1 Express helps you get started in the source, imo. I mean, at least it gives you a jumping off point, a working example... something to fiddle around with. [expressjs.com/guide.html](#) – [Wolfpack'08](#) Jul 27 '12 at 5:30