Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute
Help
Learn to edit
Community portal
Recent changes
Upload file

Tools
What links here
Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Print/export
Download as PDF
Printable version

In other projects
Wikimedia Commons
Wikibooks

Languages
Беларуская
Български
Ελληνικά
Magyar
Polski
Русский
Türkçe
Українська
中文
文A 37 more
Edit links

Node.js

From Wikipedia, the free encyclopedia

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm,[6] unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

Though `.js` is the standard filename extension for JavaScript code, the name "Node.js" doesn't refer to a particular file in this context and is merely the name of the product. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).[7]

The Node.js distributed development project was previously governed by the Node.js Foundation,[8] and has now merged with the JS Foundation to form the OpenJS Foundation, which is facilitated by the Linux Foundation's Collaborative Projects program.[9]

Corporate users of Node.js software include GoDaddy,[10] Groupon,[11] IBM,[12] LinkedIn,[13][14] Microsoft,[15][16] Netflix,[17] PayPal,[18][19] Rakuten, SAP,[20] Voxer,[21] Walmart,[22] Yahoo!,[23] and Amazon Web Services.[24]

Node.js

Original author(s)	Ryan Dahl
Developer(s)	OpenJS Foundation
Initial release	May 27, 2009; 11 years ago[1]
Stable release	15.12.0[2] / March 17, 2021; 5 days ago
Repository	github.com/nodejs/node
Written in	C, C++, JavaScript
Operating system	z/OS, Linux, macOS, Microsoft Windows, SmartOS, FreeBSD, OpenBSD, IBM AIX[3]
Type	Runtime environment
License	MIT License[4][5]
Website	nodejs.org

Contents [hide]

History [edit]

Node.js was written initially by Ryan Dahl in 2009,[25] about thirteen years after the introduction of the first server-side JavaScript environment, Netscape's LiveWire Pro Web.[26] The initial release supported only Linux and Mac OS X. Its development and maintenance was led by Dahl and later sponsored by Joyent.[27]

Ryan Dahl, creator of Node.js, in 2010

Dahl criticized the limited possibilities of the most popular web server in 2009, Apache HTTP Server, to handle a lot of concurrent connections (up to 10,000 and more) and the most common way of creating code (sequential programming), when code either blocked the entire process or implied multiple execution stacks in the case of simultaneous connections.[28]

Dahl demonstrated the project at the inaugural European JSConf on November 8, 2009.[29][30][31] Node.js combined Google's V8 JavaScript engine, an event loop, and a low-level I/O API.[32]

In January 2010, a package manager was introduced for the Node.js environment called *npm*.[33] The package manager makes it easier for programmers to publish and share source code of Node.js packages and is designed to simplify installation, updating, and uninstallation of packages.[32]

In June 2011, Microsoft and Joyent implemented a native Windows version of Node.js.[34] The first Node.js build supporting Windows was released in July 2011.

In January 2012, Dahl stepped aside, promoting coworker and *npm* creator Isaac Schlueter to manage the project.[35] In January 2014, Schlueter announced that Timothy J. Fontaine would lead the project.[36]

In December 2014, Fedor Indutny started io.js, a fork of Node.js. Due to the internal conflict over Joyent's governance, io.js was created as an open governance alternative with a separate technical committee.[37][38] Unlike Node.js,[39] the authors planned to keep io.js up-to-date with the latest releases of the Google V8 JavaScript engine.[40]

In February 2015, the intent to form a neutral Node.js Foundation was announced. By June 2015, the Node.js and io.js communities voted to work together under the Node.js Foundation.[41]

In September 2015, Node.js v0.12 and io.js v3.3 were merged back together into Node v4.0.[42] This merge brought V8 ES6 features into Node.js and a long-term support release cycle.[43] As of 2016, the io.js website recommends that developers switch back to Node.js and that no further releases of io.js are planned due to the merge.[44]

In 2019, the JS Foundation and Node.js Foundation merged to form the OpenJS Foundation.

Overview [edit]

Node.js allows the creation of Web servers and networking tools using JavaScript and a collection of "modules" that handle various core functionalities.[29][32][45][46][47] Modules are provided for file system I/O, networking (DNS, HTTP, TCP, TLS/SSL, or UDP), binary data (buffers), cryptography functions, data streams, and other core functions.[32][46][48] Node.js's modules use an API designed to reduce the complexity of writing server applications.[32][46]

JavaScript is the only language that Node.js supports natively, but many compile-to-JS languages are available.[49] As a result, Node.js applications can be written in CoffeeScript,[50] Dart, TypeScript, ClojureScript and others.

Node.js is primarily used to build network programs such as Web servers.[45] The most significant difference between Node.js and PHP is that most functions in PHP block until completion (commands only execute after previous commands finish), while Node.js functions are non-blocking (commands execute concurrently or even in parallel,[51][52] and use callbacks to signal completion or failure).[45]

Node.js is officially supported on Linux, macOS and Microsoft Windows 8.1 and Server 2012 (and later),[3] with tier 2 support for SmartOS and IBM AIX and experimental support for FreeBSD. OpenBSD also works, and LTS versions available for IBM i (AS/400).[53] The provided source code may also be built on similar operating systems to those officially supported or be modified by third parties to support others such as NonStop OS[54] and Unix servers.

Platform architecture [edit]

Node.js brings event-driven programming to web servers, enabling development of fast web servers in JavaScript.[32] Developers can create scalable servers without using threading, by using a simplified model of event-driven programming that uses callbacks to signal the completion of a task.[32] Node.js connects the ease of a scripting language (JavaScript) with the power of Unix network programming.[32]

Node.js was built on top of Google's V8 JavaScript engine since it was open-sourced under the BSD license. It is proficient with internet fundamentals such as HTTP, DNS, and TCP.[29] JavaScript was also a well-known language, making Node.js accessible to the web development community.[29]

Industry support [edit]

There are thousands of open-source libraries for Node.js, most of them hosted on the npm website. The Node.js developer community has two main mailing lists and the IRC channel #node.js on freenode. There are multiple developer conferences and events that support the Node.js community, including NodeConf, Node Interactive, and Node Summit as well as a number of regional events.

The open-source community has developed web frameworks to accelerate the development of applications. Such frameworks include Connect, Express.js, Socket.IO, Feathers.js, Koa.js, Hapi.js, Sails.js, Meteor, Derby, and many others.[32][55] Various packages have also been created for interfacing with other languages or runtime environments such as Microsoft .NET.[56]

Modern desktop IDEs provide editing and debugging features specifically for Node.js applications. Such IDEs include Atom, Brackets, JetBrains WebStorm,[57][58] Microsoft Visual Studio (with Node.js Tools for Visual Studio,[59] or TypeScript with Node definitions,[60][61][62][63]) NetBeans,[64] Nodeclipse Enide Studio[65] (Eclipse-based), and Visual Studio Code.[66][67] Certain online web-based IDEs also support Node.js, such as Codeanywhere, Codenvy, Cloud9 IDE, Koding, and the visual flow editor in Node-RED.

Node.js is supported across a number of cloud hosting platforms like Jelastic, Google Cloud Platform, AWS Elastic Beanstalk, Joyent and others.

Releases [edit]

New major releases of Node.js are cut from the GitHub master branch every six months. Even-numbered versions are cut in April and odd-numbered versions are cut in October. When a new odd version is released, the previous even version undergoes transition to Long Term Support (LTS), which gives that version 18 months of active support from the date it is designated LTS. After these 18 months expire, an LTS release receives an additional 12 months of maintenance support. An active version receives non-breaking backports of changes a few weeks after they land in the current release. A maintenance release only receives critical fixes and documentation updates.[68] The LTS Working Group manages strategy and policy in collaboration with the Technical Steering Committee of the Node.js Foundation.

Release	Status	Code name	Release date	Active LTS start	Maintenance start	Maintenance end
v0.10.x	End-of-Life		2013-03-11	-	2015-10-01	2016-10-31
v0.12.x	End-of-Life		2015-02-06	-	2016-04-01	2016-12-31
4.x	End-of-Life	Argon[68]	2015-09-08	2015-10-01	2017-04-01	2018-04-30
5.x	End-of-Life		2015-10-29	N/A		2016-06-30
6.x	End-of-Life	Boron[68]	2016-04-26	2016-10-18	2018-04-30	2019-04-30
7.x	End-of-Life		2016-10-25	N/A		2017-06-30
8.x	End-of-Life	Carbon[68]	2017-05-30	2017-10-31	2019-01-01[69]	2019-12-31
9.x	End-of-Life		2017-10-01	N/A		2018-06-30
10.x	Maintenance LTS	Dubnium[68]	2018-04-24	2018-10-30	2020-05-19	2021-04-30
11.x	End-of-Life		2018-10-23	N/A		2019-06-01
12.x	Maintenance LTS	Erbium[68]	2019-04-23	2019-10-21	2020-11-30	2022-04-30
13.x	End-of-Life		2019-10-22	N/A	2020-04-01	2020-06-01
14.x	Active LTS	Fermium[68]	2020-04-21	2020-10-27	2021-10-19	2023-04-30
15.x	**Current**		2020-10-20	N/A	2021-04-01	2021-06-01
16.x	Pending	Gallium[68]	2021-04-20	2021-10-26	2022-10-18	2024-04-30

Legend: 🟥 Old version 🟨 Older version, still maintained 🟩 **Latest version** 🟦 Future release

Technical details [edit]

Node.js is a JavaScript runtime environment that processes incoming requests in a loop, called the event loop.

Internals [edit]

Node.js uses libuv underhood to handle asynchronous events. Libuv is an abstraction layer for network and file system functionality on both Windows and POSIX-based systems such as Linux, macOS, OSS on NonStop, and Unix.

Threading [edit]

Node.js operates on a single-thread event loop, using non-blocking I/O calls, allowing it to support tens of thousands of concurrent connections without incurring the cost of thread context switching.[70] The design of sharing a single thread among all the requests that use the observer pattern is intended for building highly concurrent applications, where any function performing I/O must use a callback. To accommodate the single-threaded event loop, Node.js uses the libuv library—which, in turn, uses a fixed-sized thread pool that handles some of the non-blocking asynchronous I/O operations.[7]

A thread pool handles the execution of parallel tasks in Node.js. The main thread function call posts tasks to the shared task queue, which threads in the thread pool pull and execute. Inherently non-blocking system functions such as networking translate to kernel-side non-blocking sockets, while inherently blocking system functions such as file I/O run in a blocking way on their own threads. When a thread in the thread pool completes a task, it informs the main thread of this, which in turn, wakes up and executes the registered callback.

A downside of this single-threaded approach is that Node.js doesn't allow vertical scaling by increasing the number of CPU cores of the machine it is running on without using an additional module, such as cluster,[51] StrongLoop Process Manager,[71] or pm2.[72] However, developers can increase the default number of threads in the libuv thread pool. The server operating system (OS) is likely to distribute these threads across multiple cores.[73] Another problem is that long-lasting computations and other CPU-bound tasks freeze the entire event-loop until completion.[*citation needed*]

V8 [edit]

V8 is the JavaScript execution engine which was initially built for Google Chrome. It was then open-sourced by Google in 2008. Written in C++, V8 compiles JavaScript source code to native machine code at runtime. [7] As of 2016, it also includes Ignition, a bytecode interpreter.

Package management [edit]

npm is the pre-installed package manager for the Node.js server platform. It installs Node.js programs from the npm registry, organizing the installation and management of third-party Node.js programs. Packages in the npm registry can range from simple helper libraries such as Lodash to task runners such as Grunt.

Unified API [edit]

Node.js can be combined with a browser, a database that supports JSON data (such as Postgres,[74] MongoDB, or CouchDB) and JSON for a unified JavaScript development stack. With the adaptation of what were essentially server-side development patterns such as MVC, MVP, MVVM, etc., Node.js allows the reuse of the same model and service interface between client side and server side.

Event loop [edit]

Node.js registers with the operating system so the OS notifies it of connections and issues a callback. Within the Node.js runtime, each connection is a small heap allocation. Traditionally, relatively heavyweight OS processes or threads handled each connection. Node.js uses an event loop for scalability, instead of processes or threads.[75] In contrast to other event-driven servers, Node.js's event loop does not need to be called explicitly. Instead, callbacks are defined, and the server automatically enters the event loop at the end of the callback definition. Node.js exits the event loop when there are no further callbacks to be performed.

WebAssembly [edit]

Node.js supports WebAssembly and as of Node 14 has experimental support of WASI, the WebAssembly System Interface.

Native bindings [edit]

Node.js provides a way to make "addons" via a C-based API called N-API which can be used to produce loadable (importable) `.node` modules from source code written in C/C++.[76] The modules can be directly loaded into memory and executed from within JS environment as simple CommonJS modules. The implementation of the N-API relies on internal C/C++ Node.js and V8 objects requiring users to import (#include) Node.js specific headers into their native source code.[76] As Node.js platform constantly evolves the API compatibility is subject to changes and may get broken sometimes by a new version (as consequence modules have to be built against specific Node.js versions to work correctly). To address the issue third parties have introduced open-sourced C/C++ wrappers on top of the API that partially alleviate the problem. They simplify interfaces but as side effect they may also introduce complexity which maintainers have to deal with. Even though the core functionality of Node.js resides in a JavaScript built-in library, modules written in C++ can be used to enhance capabilities and to improve performance of applications.

In order to produce such modules one needs to have an appropriate C++ compiler and necessary headers (the latter are typically shipped with Node.js itself): gcc, clang or MSVC++.

The N-API is similar to Java Native Interface.

Project governance [edit]

In 2015, various branches of the greater Node.js community began working under the vendor-neutral Node.js Foundation. The stated purpose of the organization "is to enable widespread adoption and help accelerate development of Node.js and other related modules through an open governance model that encourages participation, technical contribution, and a framework for long-term stewardship by an ecosystem invested in Node.js' success."[77]

The Node.js Foundation Technical Steering Committee (TSC) is the technical governing body of the Node.js Foundation. The TSC is responsible for the core Node.js repo as well as dependent and adjacent projects. Generally the TSC delegates the administration of these projects to working groups or committees.[78] The LTS group that manages long term supported releases is one such group. Other current groups include Website, Streams, Build, Diagnostics, i18n, Evangelism, Docker, Addon API, Benchmarking, Post-mortem, Intl, Documentation, and Testing.[79]

In August 2017, a third of the TSC members resigned due to a dispute related to the project's code of conduct.[80]

References [edit]

1. ^ "node-v0.x-archive on GitHub". Retrieved 2 August 2014.
2. ^ "Release 2021-03-17, Version 15.12.0". Retrieved 21 March 2021.
3. ^ a b "nodejs/node". *GitHub*.
4. ^ "node/LICENSE at master". *GitHub*. Node.js Foundation. 17 September 2018. Retrieved 17 September 2018.
5. ^ "The MIT License". *Open Source Initiative*. 17 September 2018. Retrieved 17 September 2018.
6. ^ gcuomo (24 October 2013). "JavaScript Everywhere and the Three Amigos (Into the wild BLUE yonder!)". *www.ibm.com*.
7. ^ a b c Laurent Orsini (7 November 2013). "What You Need To Know About Node.js". readwrite. Archived from the original on 7 November 2013. Retrieved 22 January 2016.
8. ^ "Node.js Foundation - Node.js". Retrieved 4 July 2015.
9. ^ "Linux Foundation Collaborative Projects". Retrieved 4 July 2015.
10. ^ Why GoDaddy's Nodejitsu deal is great for Node.js, VentureBeat, 10 February 2015
11. ^ Geitgey, Adam (30 October 2013). "I-Tier: Dismantling the Monoliths". Groupon. Retrieved 30 April 2014.
12. ^ "IBM Bluemix". Retrieved 4 July 2015.
13. ^ "You'll never believe how LinkedIn built its new iPad app". VentureBeat. 2 May 2012. Retrieved 10 May 2012.
14. ^ "Blazing fast node.js: 10 performance tips from LinkedIn Mobile". Retrieved 7 April 2015.
15. ^ Baxter-Reynolds, Matthew (9 November 2011). "Here's why you should be happy that Microsoft is embracing Node.js". *The Guardian*. London. Retrieved 10 May 2012.
16. ^ "WebMatrix - Front End Web Developers take note (ASP.NET, PHP, node.js and more)". Retrieved 2 August 2014.
17. ^ Node.js in Flames 19 November 2014
18. ^ "Clash of the Titans: Releasing the Kraken, NodeJS @paypal". fluentconf.com. 28 May 2013. Retrieved 11 September 2013.
19. ^ "All such companies and their products in which Node.js is used". Retrieved 2 August 2014.
20. ^ "SAP AppBuilder". SAP. 10 March 2014. Archived from the original on 12 February 2014. Retrieved 10 October 2018.
21. ^ The Node Ahead: JavaScript leaps from browser into future, The Register, 1 March 2011
22. ^ "Why Walmart is using Node.js". VentureBeat. 24 January 2012. Retrieved 10 May 2012.
23. ^ "Why You Really Need (A) node.js development services". codebrahma. 10 January 2018. Retrieved 10 January 2018.
24. ^ "Deploy a Node.js Web App - Launch and run a highly available Node.js web application on AWS". amazon.com. Retrieved 3 October 2020.
25. ^ "About Node.js, and why you should add Node.js to your skill set?". *Training.com*. Training.com. Retrieved 23 October 2016.
26. ^ "Netscape opens intranet attack". *CNET*. Retrieved 20 April 2017.
27. ^ Ryan Dahl (9 November 2010). "Joyent and Node". Google Groups. Retrieved 5 February 2015.
28. ^ "PHP 7 vs Node.js? They Can Be Partners, Not Competitors For a Developer!". Retrieved 21 December 2016.
29. ^ a b c d Sams Teach Yourself Node.js in 24 Hours, Sams Publishing, 05-Sep-2012
30. ^ "Ryan Dahl at JSConf EU 2009".
31. ^ "Ryan Dahl at JSConf EU 2009 Video".
32. ^ a b c d e f g h i Professional Node.js: Building JavaScript Based Scalable Software, John Wiley & Sons, 01-Oct-2012
33. ^ "Earliest releases of npm". *GitHub*. Retrieved 27 July 2016.
34. ^ "Porting Node to Windows With Microsoft's Help". Retrieved 17 April 2016.
35. ^ Dahl, Ryan. "New gatekeeper". Retrieved 26 October 2013.
36. ^ Schlueter, Isaac (15 January 2014). "The Next Phase of Node.js". Retrieved 21 January 2014.
37. ^ Krill, Paul (4 December 2014). "Why io.js Decided to Fork Node.js". JavaWorld. Retrieved 15 December 2014.
38. ^ Q&A: Why io.js decided to fork Node.js, InfoWorld Tech Watch
39. ^ Ben Noordhuis (12 November 2014). "Issue 3692: function suddenly becomes undefined". V8 JavaScript Engine Issues. Retrieved 2 February 2015.
40. ^ Mikeal, Rogers (28 January 2015). "State of io.js". Retrieved 2 February 2015.
41. ^ "Node.js Foundation Advances Community Collaboration, Announces New Members and Ratified Technical Governance". Archived from the original on 24 June 2015. Retrieved 4 July 2015.
42. ^ "Node.js Foundation Combines Node.js and io.js Into Single Codebase in New Release". Retrieved 28 January 2016.
43. ^ "io.js and Node.js merge". Retrieved 27 June 2015.
44. ^ Io.js, JavaScript I/O, *"io.js has merged with the Node.js project again. There won't be any further io.js releases. All of the features in io.js are available in Node.js v4 and above."*
45. ^ a b c Node.js for PHP Developers, O'Reilly Media, Inc., 2013
46. ^ a b c Node.js Essentials, Packt Publishing, 10-Nov-2015
47. ^ Smashing Node.js: JavaScript Everywhere, John Wiley & Sons, 14-Aug-2012
48. ^ Modules, Nodejs Website
49. ^ "List of languages that compile to JS". *Github*.
50. ^ *CoffeeScript on Node.js*. O'Reilly Media, Inc. 15 April 2013. ISBN 9781449316846. Retrieved 17 May 2015.
51. ^ a b "Node.js's cluster module". Retrieved 19 October 2017.
52. ^ "Node.js's child_process module". Retrieved 19 October 2017.
53. ^ Foundation, Node js. "Installing Node.js via package manager". *Node.js*. Retrieved 19 May 2019.
54. ^ "bomBora - Node.js for NonStop". Infrasoft. Retrieved 14 August 2015.
55. ^ Express.js Guide: The Comprehensive Book on Express.js, Azat Mardan, 28-May-2014
56. ^ Edge.js bridges the gap between Node.js and .NET, TechRepublic, Tony Patton, 1 July 2014
57. ^ Node.js, WebStorm supports integration with the Node.js runtime environment, WebStorm Help
58. ^ Running and Debugging Node.js, WebStorm Help
59. ^ "Node.js Tools for Visual Studio". Retrieved 1 February 2016.
60. ^ soywiz/typescript-node-definitions TypeScript's typings for some popular node.js modules, GitHub
61. ^ DefinitelyTyped, GitHub
62. ^ The repository for high quality TypeScript type definitions Archived 3 February 2016 at the Wayback Machine
63. ^ ImageBoard, A Node.js + Express + MongoDB application built using TypeScript on the server, TypeScript Samples
64. ^ Krill, Paul (10 November 2015). "Node.js takes center stage in NetBeans 8.1". InfoWorld.
65. ^ Nodeclipse, Enide -- Node.JS development in Eclipse, Nodeclipse Website
66. ^ Hello Visual Studio Code (with NodeJS), Channel 9, Microsoft
67. ^ Node.js Applications with VS Code, Visual Studio Code
68. ^ a b c d e f g h "nodejs/Release". *GitHub*. Retrieved 3 May 2018.
69. ^ "Node 8 reschedule". Retrieved 22 January 2019.
70. ^ "Node.js w/1M concurrent connections!". *caustik's blog*.
71. ^ "StrongLoop Process Manager".
72. ^ "GitHub - Unitech/pm2: Production process manager for Node.js applications with a built-in load balancer". *GitHub*.
73. ^ Aleksander Kasiuk (22 April 2015). "On problems with threads in node.js - Future Processing".
74. ^ "PostgreSQL: Documentation: 9.4: JSON Types". *www.postgresql.org*. Retrieved 20 October 2016.
75. ^ About Node.js, Node.js Website
76. ^ a b "C++ addons | Node.js v14.12.0 Documentation". *nodejs.org*. Retrieved 26 September 2020.
77. ^ Foundation, Node.js. "Node.js Foundation | Node.js". *nodejs.org*. Retrieved 8 March 2017.
78. ^ "nodejs/TSC". *GitHub*. Retrieved 8 March 2017.
79. ^ Foundation, Node.js. "Working Groups | Node.js". *nodejs.org*. Retrieved 8 March 2017.
80. ^ Whittaker, Zack, *After governance breakdown, Node.js leaders fight for its survival*, ZDNet, 25 August 2017

Further reading [edit]

- Hughes-Croucher, Tom; Wilson, Mike (April 2012), *Up and Running with Node.js*, O'Reilly Media, ISBN 978-1-4493-9858-3
- Ornbo, George (September 2012), *Sams Teach Yourself Node.js in 24 Hours*, SAMS Publishing, ISBN 978-0-672-33595-2
- Teixeira, Pedro (October 2012), *Professional Node.js*, John Wiley & Sons, ISBN 978-1-118-22754-1
- Randal L. Schwartz and Aaron Newcomb (9 January 2013). "Episode 237: Node.js". *twit.tv/show/floss-weekly* (Podcast). TWiT.tv. Event occurs at 1:08:13. Retrieved 9 January 2013.
- Gackenheimer, Cory (October 2013), *Node.js Recipes: A Problem-Solution Approach*, Apress, ISBN 978-1-4302-6058-5

External links [edit]

- Official website
- GitHub Repository

Wikimedia Commons has media related to *Node.js*.

This page was last edited on 18 March 2021, at 10:02 (UTC).