



Node.js What's Next?

Michael Dawson

Node.js lead for Red Hat and IBM

About Michael Dawson

Node.js lead for Red Hat and IBM

Active Node.js community member

Node.js Collaborator, Node.js Technical Steering Committee,

Community Committee member

Active in a number of Working group(s)

Active OpenJS Foundation member

Voting Cross Project Council Member

Community Director 2020-2021

Twitter: @mhdawson1

GitHub: @mhdawson

Linkedin: <https://www.linkedin.com/in/michael-dawson-6051282>



Agenda

- How do I track “what’s” next?
- Node.js Releases
- Interesting New Features
- Strategic Initiatives
- What are the teams and Working groups up to?
- What’s new in the OpenJS Foundation?
- Getting Involved

Agenda

- Node.js project has no formal roadmap!
 - No single corporate sponsor
 - Decentralized
 - Things make it into release when ready
- But!
 - Still longer term efforts and planning



Agenda

- Releases
- Twitter
- GitHub Fire Hose
- Working Groups/Teams
- Strategic Initiatives
- OpenJS Foundation



nodejs.org

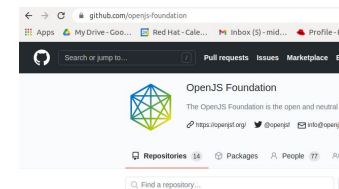
github.com/nodejs - 187 Repositories

github.com/pkgjs - 13 Repositories

<https://nodejs.org/calendar>

github.com/nodejs/TSC/blob/master/Strategic-Initiatives.md

github.com/nodejs/community-committee/blob/master/STRATEGIC-INITIATIVES.md

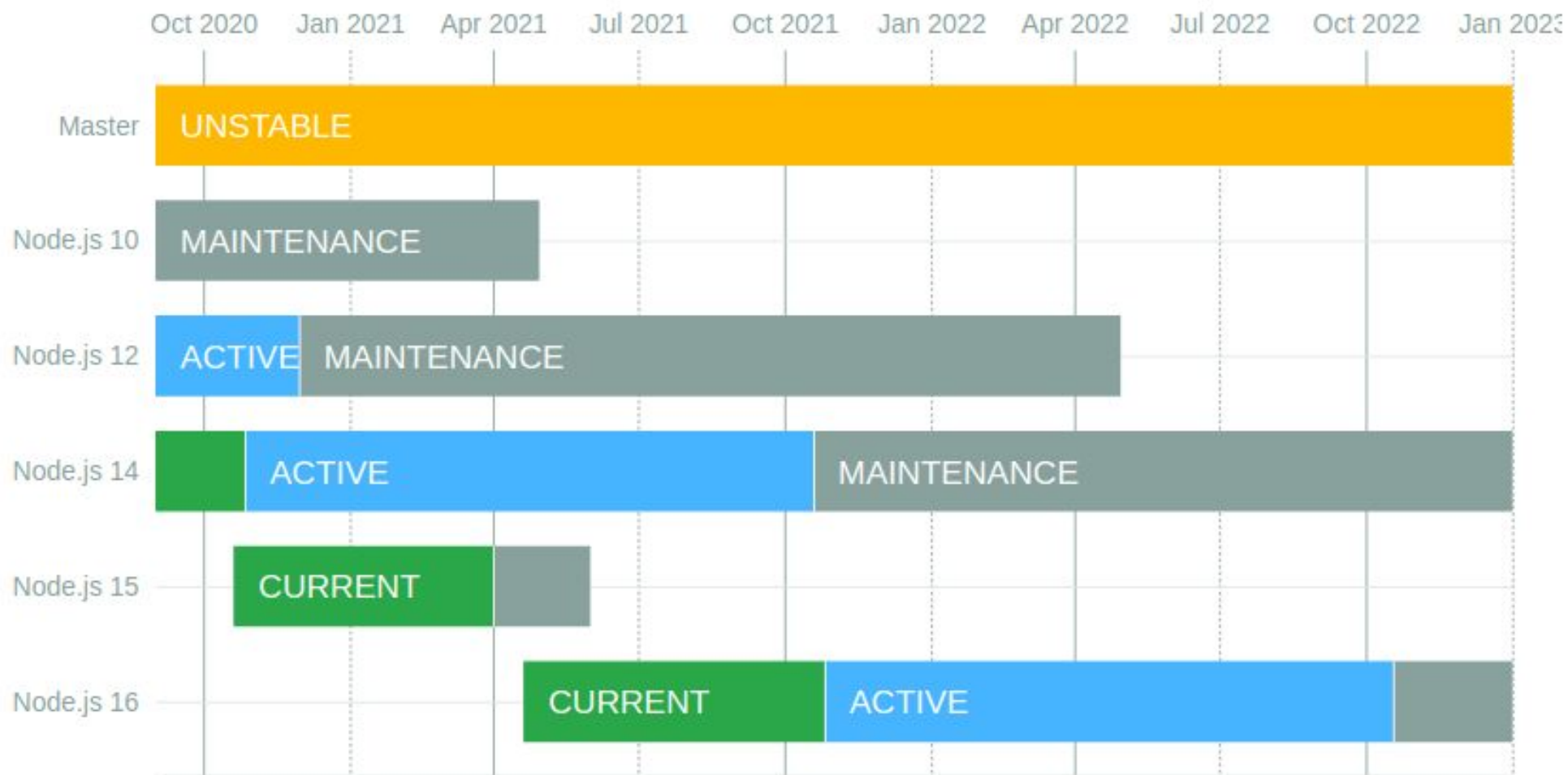


github.com/openjs-foundation
openjsf.org

Node.js Releases


- Release Types
 - Canary
 - Nightlies
 - Current
 - Every 6 months
 - Even releases promoted to LTS
- LTS
 - Every 12 months
 - 30 Months support (12 active, 18 maintenance)

Node.js Releases



source: github.com/nodejs/release

Node.js Releases

☐  **Release plan - v15.x Current** Release-agenda

#621 opened on 19 Oct by BethGriggs

2020-11-16	v15.2.1 (security)	@targos
2020-11-24	v15.x	@codebytere
2020-12-08	v15.x	@danielleadams
2020-12-22	v15.x	@targos
2021-01-05		
2021-01-19		
2021-02-02		

☐  **Release plan - v14.x Active LTS** Release-agenda

#567 opened on 14 Apr by BethGriggs

☐  **Release plan - v12.x Active LTS** Release-agenda

#494 opened on 21 Oct 2019 by targos

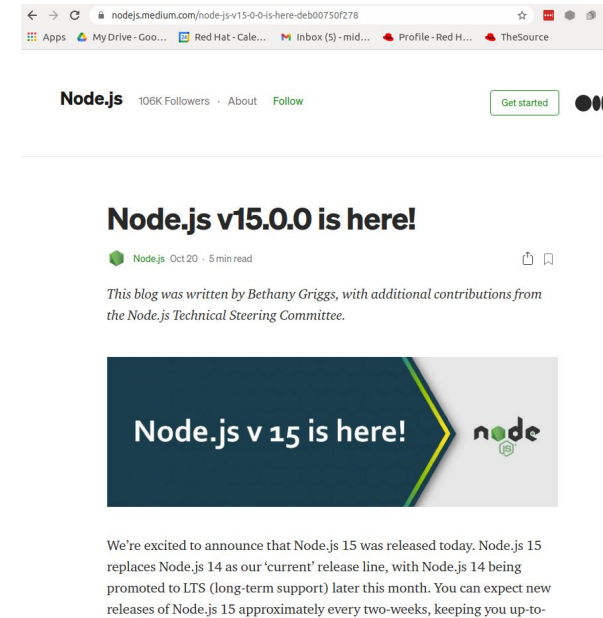
☐  **Release plan - v10.x Maintenance** Release-agenda

#504 opened on 18 Nov 2019 by BethGriggs

github.com/nodejs/release

Interesting New Features

- Why do “Majors” seem so boring?
- Interesting features often SemVer Minor
- Features flow backwards quickly
- By the time we have a Major they are “old” news
- Still important -> Transition to LTS !



nodejs.medium.com/node-js-v15-0-0-is-here-deb00750f278

Announcing core Node.js support for ECMAScript modules



Node.js Nov 21, 2019 · 5 min read

This article was contributed by the Node.js Module Team.

Node.js 13.2.0 ships support for ECMAScript modules, known for their `import` and `export` statements. This support was previously behind the `--experimental-`

nodejs.medium.com/announcing-core-node-js-support-for-ecmascript-modules-c5d6dc29b663

Interesting New Features

- ES Modules
- Throw on Unhandled Rejections
- Full ICU by default
- Diagnostic Report
- N-API V7 (covered in section on Working Groups)
- Experimental
 - Async Local Storage APIs
 - Abort Controller
 - QUIC

ES6 Modules

CommonJS

```
module.exports = function() {  
  console.log('Hello');  
}
```

```
const test =  
require('./helloTest.js');  
test();
```



ES6

```
export function test() {  
  console.log('Hello ES6');  
}
```

```
import { test } from  
'./helloTest.mjs';  
test();
```

ES6 Modules

CommonJS

```
module.exports = function() {  
  console.log('Hello');  
}
```

```
const test =  
require('./helloTest.js');  
test();
```

ES6

```
export function test() {  
  console.log('Hello ES6');  
}
```

```
import { test } from  
'./helloTest.mjs';  
test();
```

Spent a good amount of time “baking” and working on Improvements in Ergonomics

```
Package.json  →  {  
                  "name": "test",  
                  "type": "module"  
                }  →  import { test } from  
                      './helloTest2.js';  
                      test();
```

<https://nodejs.org/api/esm.html>

ES6 Modules

CommonJS

```
module.exports = function() {  
  console.log('Hello');  
}
```

```
const test =  
require('./helloTest.js');  
test();
```

ES6

```
export function test() {  
  console.log('Hello ES6');  
}
```

```
import { test } from  
'./helloTest.mjs';  
test();
```


Spent a good amount of time “baking” and working on Improvements in Ergonomics

```
Package.json → {  
  "name": "test",  
  "type": "module"  
} → import { test } from  
      './helloTest2.js';  
      test();
```

<https://nodejs.org/api/esm.html>

Now Stable in master (Nov 18)

doc: mark ECMAScript modules implementation as stable #35781

 Closed  guybedford wants to merge 1 commit into master from stabilize-modules

 Conversation 21  Commits 1  Checks 14  Files changed 1



guybedford commented on 23 Oct • edited

Member  ...

Per discussion at the last meeting this is a PR to mark the entire ECMAScript modules implementation as stable while keeping the individual subfeatures unstable as appropriate.

Reviewers

 mcollins

 fthinkel

Throw on Unhandled Rejections

```
new Promise((resolve, reject) => {  
  reject('error');  
});
```

(node:31727) UnhandledPromiseRejectionWarning: error
(node:31727) UnhandledPromiseRejectionWarning: Unhandled promise
(node:31727) [DEP0018] DeprecationWarning: Unhandled promise re

Previous Versions

```
node:internal/process/promises:218  
  triggerUncaughtException(err, true /* fromPromise */);  
  ^  
  
[UnhandledPromiseRejection: This error originated either by throwing inside of a  
  code: 'ERR_UNHANDLED_REJECTION'  
]
```

15+

Throw on Unhandled Rejections

Read more in:

<https://developer.ibm.com/languages/node-js/blogs/nodejs-15-release-blog/>

```
new Promise((resolve, reject) => {  
  reject('error');  
});
```

```
(node:31727) UnhandledPromiseRejectionWarning: error  
(node:31727) UnhandledPromiseRejectionWarning: Unhandled promise  
(node:31727) [DEP0018] DeprecationWarning: Unhandled promise re
```

Previous Versions

```
node:internal/process/promises:218  
  triggerUncaughtException(err, true /* fromPromise */);  
  ^  
  
[UnhandledPromiseRejection: This error originated either by throwing inside of a  
  code: 'ERR_UNHANDLED_REJECTION'  
]
```

15+

Fixing

```
NODE_OPTIONS=- -unhandled-rejections=warn
```

```
new Promise((resolve, reject) => {  
  reject('error');  
}).catch((error) => {});
```

```
process.on('unhandledRejection', (reason, promise) => {  
  // do something  
});
```

Full ICU by default

- 14.x -> first LTS with full ICU

```
console.log(new Intl.DateTimeFormat('es',{month:'long'}).format(new Date(9E8)));  
console.log(new Date(0).toLocaleString("el",{month:"long"}));  
console.log(new Date(157177E7).toLocaleString("zh",{year:"numeric",day:"numeric",month:"long"}));
```

```
bash-4.2$ node test2.js  
January  
December  
October 22, 2019
```



```
bash-4.2$ node test2.js  
enero  
Δεκεμβρίου  
2019年10月22日
```


Diagnostic Report

- 14.x -> first LTS with Diagnostic report as stable

- JSON, easily consumable

- Key FDDC info

- Command line args
- Process info
- GC/Heap info
- Component versions
- Javascript and Native stacks
- Environment variables

- Command line

- --diagnostic-report-uncaught-exception
- --diagnostic-report-on-signal
- --diagnostic-report-on-fatalerror

- Programatically `process.report.writeReport();`

<https://nodejs.org/dist/latest-v11.x/docs/api/report.html>

<https://developer.ibm.com/articles/easily-identify-problems-in-your-nodejs-apps-with-diagnostic-report/>

<https://developer.ibm.com/technologies/node-js/articles/introducing-report-toolkit-for-nodejs-diagnostic-reports>

```
dumpEventTime : 2018-12-21 00:00:11 ,
"dumpEventTimeStamp": "1545371411331",
"processId": 8974,
"cwd": "/home/nodeuser/project/node",
"commandLine": [
  "/home/nodeuser/project/node/out/Release/node",
  "--experimental-report",
  "--diagnostic-report-uncaught-exception",
  "/home/nodeuser/project/node/test/report/test-exception.js",
  "child"
],
"nodejsVersion": "v12.0.0-pre",
"libcVersionRuntime": "2.17",
"libcVersionCompiler": "2.17",
"wordSize": "64 bit",
"arch": "x64",
"platform": "linux",
"componentVersions": {
  "node": "12.0.0-pre",
  "v8": "7.1.302.28-node.5",
  "uv": "1.24.1",
  "zlib": "1.2.11",
  "ares": "1.15.0",
  "modules": "68",
  "nghttp2": "1.34.0",
  "napi": "3",
  "llhttp": "1.0.1",
  "http_parser": "2.8.0",
  "openssl": "1.1.0j"
},
"release": {
  "name": "node"
},
"osName": "Linux",
"osRelease": "3.10.0-862.el7.x86_64",
"osVersion": "#1 SMP Wed Mar 21 18:14:51 EDT 2018",
"osMachine": "x86_64",
"host": "test_machine"
},
"javascriptStack": {
  "message": "Error: *** test-exception.js: throwing uncaught Error",
  "stack": [
    "at myException (/home/nodeuser/project/node/test/report/test-excep",
    "at Object.<anonymous> (/home/nodeuser/project/node/test/report/tes",
    "at Module._compile (internal/modules/cjs/loader.js:718:30)",
```

Async Local Storage (Experimental)

- Maintain state across async calls
- Important for distributed transaction tracing

```
const AsyncLocalStorage = require('async_hooks').AsyncLocalStorage;
const asyncLocal1 = new AsyncLocalStorage();

function logData() {
  console.log(asyncLocal1.getStore(['url']));
}

function accept(request, response) {
  asyncLocal1.getStore()['url'] = request.url;
  setTimeout(logData, 100);
};

asyncLocal1.run(new Object(), accept.bind(null, { url: 'url1'}, {}));
setTimeout(() => { asyncLocal1.run(new Object(), accept.bind(null, { url: 'url2'}, {})) },
  500);

asyncLocal1.run(new Object(), accept.bind(null, { url: 'url1'}, {}));
--> asyncLocal1.getStore()
--> store1
setTimeout(() => { asyncLocal1.run(new Object(), accept.bind(null, { url: 'url2'}, {})) }, 500);
--> asyncLocal1.getStore()
--> store2
```

- **Class: AsyncLocalStorage**
 - `new AsyncLocalStorage()`
 - `asyncLocalStorage.disable()`
 - `asyncLocalStorage.getStore()`
 - `asyncLocalStorage.enterWith(store)`
 - `asyncLocalStorage.run(store, callback[, ...args])`
 - `asyncLocalStorage.exit(callback[, ...args])`

Abort Controller (Experimental)

- Cancellation/Aborting long running promise-based APIs

- Global objects
 - Class: `AbortController`
 - `abortController.abort()`
 - `abortController.signal`
 - Class: `AbortSignal`
 - Event: `'abort'`
 - `abortSignal.aborted`
 - `abortSignal.onabort`

nodejs.org/api/all.html#globals_class_abortcontroller

QUIC (Experimental)

- Compile time option `--experimental-quic`
- Early work on QUIC integration

Strategic Initiatives

Current Initiatives

Initiative	Champion	Links
Core Promise APIs	Matteo Collina	
V8 Currency	Michaël Zasso	
QUIC / HTTP3	James M Snell	https://github.com/nodejs/quic
Startup performance	Joyee Cheung	https://github.com/nodejs/node/issues/17058 https://github.com/nodejs/node/issues/21563
Build resources	Michael Dawson	https://github.com/nodejs/build/issues/1154#issuecomment-448418977
Future of Build Toolchain	Mary Marchini	https://github.com/nodejs/TSC/issues/901 , https://github.com/nodejs/build-toolchain-next

github.com/nodejs/TSC/blob/master/Strategic-Initiatives.md

Strategic Initiatives

Current Initiatives

Initiative	Champion(s)	CommComm Liason	Link	OKR
i18n	obensource		nodejs/i18n	
Mentorship	DavidGuttman		nodejs/mentorship	
Node.js Collection	waleedashraf		nodejs/nodejs-collection	
Examples	bnb		nodejs/examples	
Outreach	AhmadAwais		nodejs/outreach	
Website Redesign	designmoreweb	obensource	nodejs/nodejs.dev	nodejs.dev/objectives-key-results

github.com/nodejs/community-committee

What are the teams and Working groups up to?

- **N-API**
- **Diagnostics**
- **Next-10**
- **Package Maintenance**
- **Web-server-frameworks**
- Build
- Website redesign
- l18n
- Release
- Ecosystem Security
- Moderation
- Tooling
- Mentorship

N-API - Modern Native Addons

- N-API in core and node-addon-api module

github.com/nodejs/abi-stable-node

github.com/nodejs/node-addon-api

N-API 4	Thread-safe Functions
N-API 5	Date objects Attach finalizer to object
N-API 6	BigInts Set Instance Data Get All Property Names
N-API 7	Detached ArrayBuffers

± Weekly Downloads

2,946,733



- Context Sensitivity and Workers
- Building
 - make in addition to node-gyp
 - node-pre-gyp, prebuild

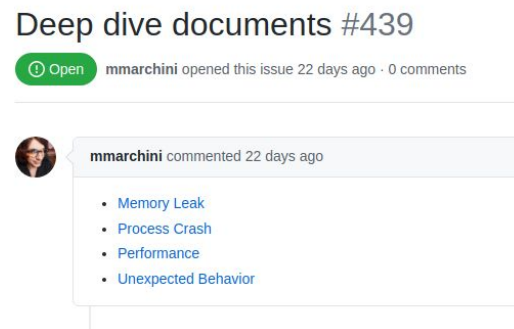
Many thanks to [Chengzhong Wu](#) for his contribution to N-API and other parts of Node.js

nodejs.medium.com/building-modern-native-add-ons-for-node-js-in-2020-cd3992c68e0

Diagnostics

- Diagnostic best practices

<https://github.com/nodejs/diagnostics>



- Context Local Storage
 - Covered earlier in interesting new features
- Diagnostic Channel
 - github.com/nodejs/diagnostics/issues/180
 - <https://github.com/nodejs/node/pull/34895>
- What about Async Hooks ?

Async hooks

Stability: 1 - Experimental

Source Code: `lib/async_hooks.js`

The `async_hooks` module provides an API to track asyn

Next-10

- Strategic directions for the next 10 years of Node.js (Building on the successful first 10)
- Technical Values – github.com/nodejs/node/blob/master/doc/guides/technical-values.md

Values and priority level

- Priority 1 - Developer experience
- Priority 2 - Stability
- Priority 3 - Operational qualities
- Priority 4 - Node.js maintainer experience
- Priority 5 - Up to date technology and APIs

- Constituencies github.com/nodejs/next-10/blob/a6ad76b9707c0eb309ad92f4a486cca8adacf6da/CONSTITUENCIES.md

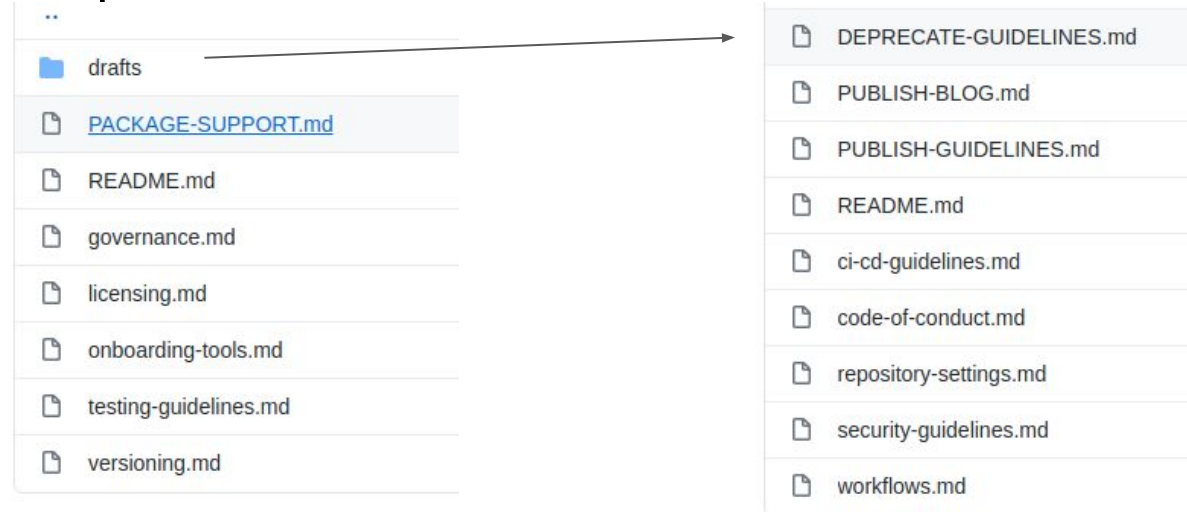
- Direct end users
 - Users who run tools themselves (install Node.js, run tool, deal with errors)
- Application operators
 - Users who interact with existing, running applications (regardless if they wrote the application or not)
- Application Developers
 - Front-end tool consumers
 - Back-end server authors
 - Hobby developers
 - Professional developers
 - Tool authors
- Library/package authors
 - Users who write libraries and packages to be included on other applications
- Node.js core maintainers
 - Developers working directly on [nodejs/node](https://github.com/nodejs/node)
 - Individuals participating in Working Groups
- Organizations with investments in Node.js (eg: Enterprises, Government bodies, startups, non-profits)

Needs of the Node.js Constituencies

Need	Users	Ops	AppDev	LibDev	Orgs	NodeMaint
Good understanding of the direction of the project	X	X	X	X	X	X
Ability to affect the direction of the project	X	X	X	X	X	X
Consumable APIs and docs	X		X	X		X
Predictable and Stable Releases	X	X	X	X	X	
Innovation at a consumable pace	X		X	X	X	X
Easy Installation	X					
Easy issue reporting, resolution and collaboration	X	X	X	X	X	
Broad Deployment platform support	X	X			X	
Broad Desktop platform support	X		X	X		X
Consistent and intuitive Error handling	X	X	X	X		
Runtime Diagnostic Tooling	X	X				
Development time Diagnostic Tooling			X	X		
Relevant APIs in core			X	X		
Module/dependency info and management	X	X	X	X		
Ways to fund their work				X		X
Ability to assess impact of changes they make				X		X
Good security and CVE practices in the project	X	X	X	X	X	
Better CVE management in the ecosystem	X	X	X	X	X	
Good CI infrastructure and experience in the project						X
Supportive Collaborators and Environment in the project						X
Better ways to build consensus in the project						X
Easy contribution workflow					X	X

<https://github.com/nodejs/next-10/blob/a6ad76b9707c0eb309ad92f4a486cca8adacf6da/CONSTITUENCY-NEEDS.md>

- Best practices



- Communication Gap - package-support

nodejs.medium.com/node-js-package-maintenance-bridging-the-gap-between-maintainers-and-consumers-9b3f154b0847

- Inter-package dependency testing - github.com/pkgjs/wiby

- Triage

Package Support

github.com/nodejs/package-maintenance/blob/main/docs/PACKAGE-SUPPORT.md

```
"support": {
  "versions": [
    {
      "version": "*",
      "target": {
        "node": "none"
      },
      "response": {
        "type": "time-permitting",
        "paid": false,
        "contact": {
          "name": "Volunteers",
          "url": "https://github.com/myproject"
        }
      },
      "backing": {
        "hobby": "https://github.com/myproject"
      }
    }
  ]
}
```

Value	
none	Don't expect
time-permitting	The maintainer
regular-x	There are
24-7	There are

abandoned						deprecated or no longer maintained
none						Use at your own risk, no active support. May or may not work for a given Node.js version
all	✓	✓	✓	✓	...11.12.13.14.15	The package is maintained for versions of Node.js including both LTS and non-LTS releases regardless of whether they are EOL or not. It may be necessary to accept semver-major level (ie. breaking) changes into that application in order to receive essential fixes. Documentation for the package will include the non-LTS releases for which the package is still maintained (some maintainers support as far back as 0.6)
lts		✓	✓		10.12	The package is maintained for the Node.js LTS releases (both in Active and Maintenance status). Anyone creating an application using an LTS version of Node.js and using the latest major version of LTS adopting packages will not have to accept semver-major level (ie. breaking) changes into that application in order to receive essential fixes. Full details are available here
active	✓	✓			10.12.14.15	Current release and active LTS releases
lts_active		✓ (all)			10.12	All releases in active LTS status. There may be more than one LTS release in active maintenance at a given point in time
lts_latest		✓ (latest)			12	The package is maintained only for the Latest LTS Node.js version. You will be required to update to the latest LTS Node.js version in order to ensure you can use new versions/get security fixes
supported	✓	✓	✓		10.12.14.15	Node.js versions which are not EOL

Value	Description
none	There is nobody backing this package.
hobby	The single maintainer maintains the package for fun, does not get any support
npm-funding	The project is asking for funding and can be funded through the links provided in package.json.
bounty	The package is maintained through the use of a bounty service.
project	The package is maintained under the auspices of a larger project (ex Node.js)
foundation	The package is maintained and supported under the auspices of a Foundation
company	The package is maintained and supported by a corporate entity but may not offer commercial offerings.
commercial	The package is maintained and supported by a corporate entity as part of a commercial offering.
paid-support	The package is maintained and supported through paid support contracts
freemium	Basic version of the package is provided for free, premium version is available
donations	The project can be funded by any donations.

Package Support

```
npx @pkgjs/support show
```

```
npx @pkgjs/support validate
```

```
user1@user1-VirtualBox:~/newpull/node-addon-api$ npx @pkgjs/support validate
Error in support JSON
TYPE should be object

> 1 | {"versions":[{"version":"*","target":{"node":"active"},"response":{"type":
:"time-permitting","paid":false,"contact":{"name":"node-addon-api team","url":"
https://github.com/nodejs/node-addon-api/issues"}},{"backing":[{"project":"https
://github.com/nodejs"}, {"foundation":"https://openjsf.org/"}]}]}
```

@pkgjs/support

0.0.3 • Public • Published 2 months ago

Readme

Explore BETA

8

Package Support

npm v0.0.3 downloads 185/month code style standard Test passing

support backing HOBBY support target LTS support response best-effort

github.com/pkgjs/support

Apps My Drive - Goo... Red Hat - Cale... Inbox (5) - mid...

Search or jump to... Pull requests Issues

pkgjs / support

<> Code Issues 4 Pull requests 3 Actions

master 2 branches 2 tags

mhdawson doc: change response to time-permitting (#2

.github/workflows add: schema i

bin add: schema i

doc Fix simple type

<https://github.com/pkgjs/support>

- Future of HTTP
- undici – github.com/nodejs/undici

- URL's on incoming requests #71
- discuss the future of HTTP client and Undici #66
- Future of HTTP High Level API Design #60
- Future of HTTP Technical Plan #55
- Future of HTTP Organization Plan #54

What's new in the OpenJS Foundation?



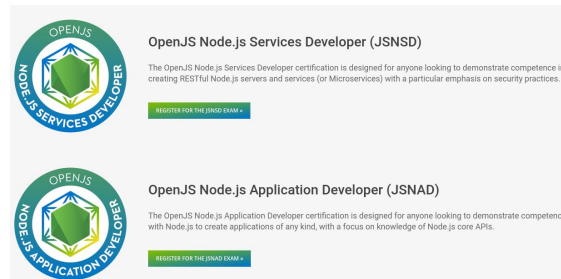
openjsf.org/

- New Projects
 - Electron
 - Amp
 - Fastify
- Collaboration Network



openjsf.org/blog/2020/11/12/introducing-the-openjs-collaboration-network/

- Certifications



Now Available in Chinese !

Many Thanks to : [Khaidi Chu](#) from the Node.js infrastructure team at Alibaba.

training.linuxfoundation.org/announcements/openjs-node-js-certification-exams-now-available-in-chinese/

So What's Next?



It's what YOU make it!

Meet us in GitHub


A vertical red bar on the left side of the slide contains a complex, stylized graphic. It features various icons: a cloud with a keyhole, a database cylinder, a server rack, a person silhouette, and several arrows pointing in different directions. Some elements are in a darker red shade than others, creating a layered effect.

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

 [linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)

 [facebook.com/redhatinc](https://www.facebook.com/redhatinc)

 [youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)

 twitter.com/RedHat



Copyright and Trademarks

© Red Hat. All Rights Reserved

Red Hat, the Red Hat logos are trademarks or registered trademarks of Red Hat

IBM, the IBM logo, ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml

Node.js is an official trademark of Joyent. IBM SDK for Node.js is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

Java, JavaScript and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

npm is a trademark of npm, Inc.

Other trademarks or logos are owned by their respective owners.