Node.js What's Next?

Michael Dawson
IBM Community Lead for Node.js

About Michael Dawson
IBM Community Lead for Node.js

- Active Node.js community member
  - Collaborator
  - Node.js Technical Steering Committee TSC Chair
  - Community Committee member
  - Working group(s) member/leadership

- Twitter: @mhdawson1
- GitHub: @mhdawson
- Linkedin: https://www.linkedin.com/in/michael-dawson-6051282

Overview

- Learn how to fish (why do I need to fish?)
- Survey of what new/next
  - Features
  - Working Groups/Teams
  - Strategic Initiatives
- Wrap-up and Questions

Why do I need to fish?

- 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

2019 – Node.js Core in Numbers

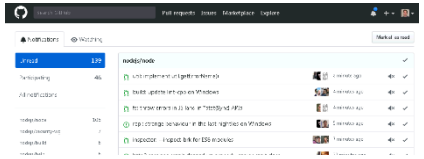
- 2,541 contributors
- 3000+ Pull requests
- 1400+ issues opened
- 40+ releases

Go Fishing, How to track What's Next

- Releases

https://nodejs.org

- Twitter
- GitHub Fire Hose



- Working Groups/Teams

https://github.com/nodejs
https://nodejs.org/calendar

- Strategic Initiatives

https://github.com/nodejs/TSC/blob/master/Strategic-Initiatives.md
https://github.com/nodejs/community-committee/blob/master/STRATEGIC-INITIATIVES.md

- OpenJS Foundation

https://openjsf.org/
https://github.com/openjs-foundation/standards

Releases - Process

Release Types

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

‘Notable Changes’ in release notes is good way to see what’s coming

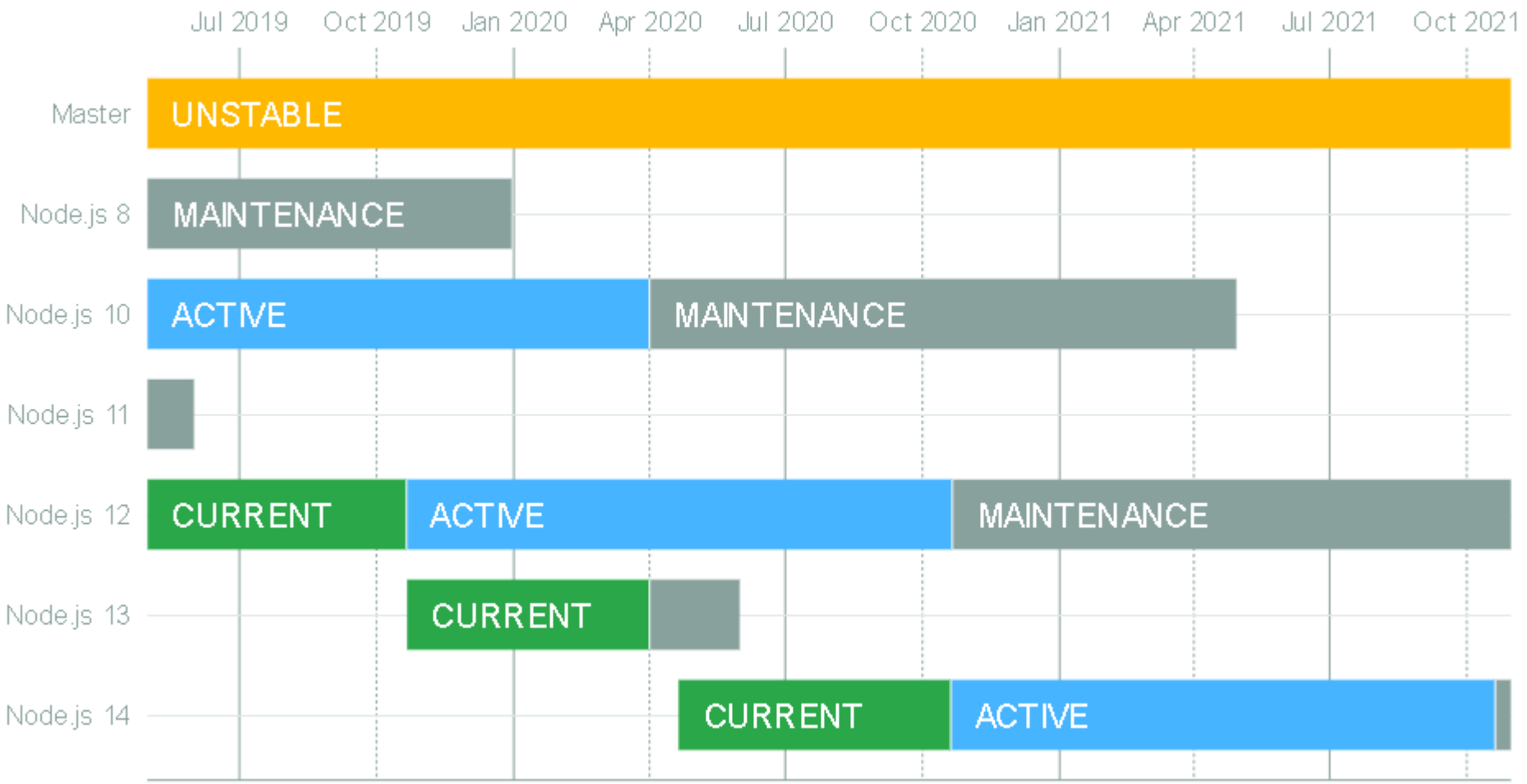
https://github.com/nodejs/node/blob/master/CHANGELOG.md

2019-01-29, Version 10.15.1 'Dubnium' (LTS), @codebytere

Notable Changes

- doc:
  - add oyyd to collaborators (Ouyang Yadong) #24300
- tls:
  - throw if protocol too long (Andre Jodat-Danbrani) #23606

Releases - Schedule for 2020



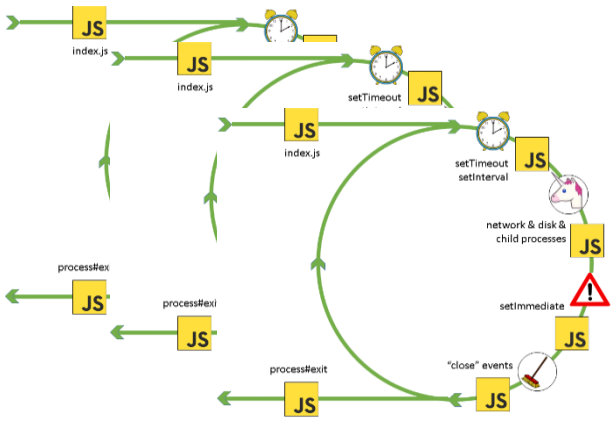
source: https://github.com/nodejs/release

Survey of what’s New/Next

- Features
- Teams/Working Groups
- Strategic Initiatives

Features – Workers – Now Stable in 12.x

- Very **similar to Web Workers**
- Each thread is **separate JavaScript environment**
- **Message-based** data exchange:
  - Object cloning via structure clone algorithm
  - Handoff via `ArrayBuffer` or `MessagePort`
  - Memory sharing via `SharedArrayBuffer` and `Atomics`
- **Limitations**
  - Cannot transfer file handles
  - Initial overhead; re-use or pooling recommended




```
// app.js
const workers = require('worker_threads');

if (workers.isMainThread) {
  const worker1 = new workers.Worker(__filename, {foo: 'bar'});
  const worker2 = new workers.Worker(__filename, {baz: 'quux'});

  worker1.on('message', message => {
    console.log(`parent:${message}`);
  });

  worker2.on('message', message => {
    console.log(`parent:${message}`);
  });

  worker2.postMessage('wakeup');
} else {
  workers.parentPort.postMessage('running');
  workers.parentPort.on('message', message => {
    console.log(`worker:${message}:${workers.threadId}`);
    workers.parentPort.postMessage(message);
  });
}

// $ node app.js
// parent:running
// parent:running
// worker:wakeup:2
// parent:wakeup
```

Http2 – stable since 10.x (ok not really new)

– HTTP/2

- Is a **binary** protocol
- Is **fully multiplexed**, instead of ordered & blocking
- Achieves **parallelism** using a single connection
- Uses **header compression** to reduce overhead
- Allows servers to “**push**” responses proactively into client caches

– Node.js provides a **core** API and a **compatibility** API

– What’s next -> QUIC

```
const http2 = require('http2');
const fs = require('fs');
const {
  HTTP2_HEADER_STATUS,
  HTTP2_HEADER_CONTENT_TYPE
} = http2.constants;

// core API
http2
  .createSecureServer({
    key: fs.readFileSync('/path/to/key.pem'),
    cert: fs.readFileSync('/path/to/cert.pem')
  })
  .on('stream', stream => {
    stream.respond({
      [HTTP2_HEADER_CONTENT_TYPE]: 'text/html',
      [HTTP2_HEADER_STATUS]: 200
     });
    stream.end('<h1>Hello World</h1>');
  })
  .listen(8443);

// compatibility API
http2
  .createSecureServer(
    {
      key: fs.readFileSync('/path/to/key.pem'),
      cert: fs.readFileSync('/path/to/cert.pem')
    },
    (req, res) => {
      res.writeHead(200, {'Content-Type': 'text/html'});
      res.end('<h1>Hello World</h1>');
    }
  )
 .listen(9443);
```

Features - Diagnostic Reports - Experimental

- Released in **Node.js v11.8.0**
- Usable **via flag only** `--experimental-report`
  - `--diagnostic-report-directory=directory`
  - `--diagnostic-report-filename=filename`
  - `--diagnostic-report-on-fatalerror`
  - `--diagnostic-report-on-signal`
  - `--diagnostic-report-signal=signal`
  - `--diagnostic-report-uncaught-exception`
  - `--diagnostic-report-verbose`
- **JSON** output; see example at https://nodejs.org/docs/latest/api/report.html

```
// automatic trigger
process.report.setDiagnosticReportOptions({
  events: ['exception', 'fatalerror', 'signal'],
  signal: 'SIGUSR2',
  filename: 'myreport.json',
  path: '/home/nodeuser',
  verbose: true
});

// manual trigger
try {
  process.chdir('/non-existent-path');
} catch (err) {
  process.report.triggerReport(err);
}

// custom handling
const report = process.report.getReport(
  new Error('custom error')
);
console.log(report); // JSON string
```

Introducing report-toolkit

https://github.com/ibm/report-toolkit

A tool for processing & analyzing Diagnostic Reports

- CLI tool
- Programmable API
- See https://ibm.github.io/report-toolkit for documentation (WIP)
- *Alpha!*

Redact – sensitive info
Inspect – ESLint-like rules
Diff
Transform

Internationalization

- Node has had Intl for a long time, but lacked language data
  - Need to add data - https://www.npmjs.com/package/full-icu

- Node 13.x brings full-icu data bundled in
  - Collation
  - Date/Time and Number formatting
  - Upper/Lower Casing

```
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"}));
```

Node.js 13.x and later

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

Node.js 12 and before

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

Locales: es = Spanish, el = Greek, zh = chineese

OpenSSL 1.1.1

- TLS 1.3
  - Enabled by default in12.x and above
  - Enabled in 10.x with --tls-max-v1.3
  - https://developer.ibm.com/technologies/node-js/blogs/migrating-to-tls13-in-nodejs/
  - https://developer.ibm.com/blogs/openssl-111-has-landed-in-nodejs-master-and-why-its-important-for-nodejs-lts-releases/
- No Community FIPs
  - Node.js went out of service Dec 31 2019
  - OpenSSL plans for FIPs on OpenSSL 3

Source Maps

- 2019 State of JavaScript Survey, ~60% using alternate flavor of JavaScript
  - Up from ~21% in 2016

- Source Maps V3 - https://sourcemaps.info/spec.html
  - Generate source includes special comment: //# sourceMappingURL=mytest.js.map

- Node.js v12.11.0 – NODE_V8_COVERAGE
  - Load/cache source maps when available

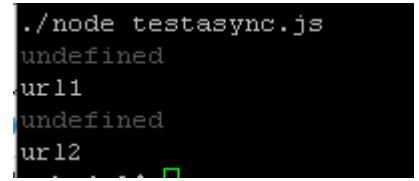
- Node.js v12.12.0 - --enable-source-maps (Experimental)
  - Better stack trace on exception

- Still more places in Node.js to update

Context Local Storage - Experimental

- Analog to ThreadLocal in other languages
- Previously user-land implementations -> ex https://www.npmjs.com/package/continuation-local-storage
- Uses
  - Async Hooks (experimental, concerns about API surface)
  - Hope is to get API we can make non-experimental sooner

- AsyncLocalStorage
  - run(callback)
  - runSyncAndReturn()
  - getStore()
  - get/set on store
  - exit(callback)
  - exitSyncAndReturn()
  - disable



Pointer Compression

- Pointers use a lot of memory in the heap
  - Potentially save 50-60%

- Master -> https://github.com/nodejs/node/pull/30463

- Project still needs to figure out how we might deliver
  - Current limit is 4G
  - Build only flag
    - More complex build
    - Increased download size

WASI – Experimental

- Web Assembly – WASM - https://webassembly.org/
  - Compile other languages (C/C++/Rust etc.) to portable target
  - WASM support in Node.js through V8

- Web Assembly System Interface WASI (https://github.com/WebAssembly/WASI)
  - adds interaction with conceptual OS
  - Initial PR for Node.js support https://github.com/nodejs/node/pull/30258

- WASI provides methods like
  - __wasi_fd_read()
  - __wasi_clock_time_get()
  - __wasi_environ_get()
  - etc.
- Language implementation maps standard functions (for example libc functions to WASI functions)

Node.js 12 - V8 Updated to V8 7.7

- Faster `await`
- Faster `JSON.parse()`
- Better Async Stack Traces

https://v8.dev/blog

from the V8 Project

Teams and Working Groups

Teams/Working Groups - Package-maintenance

Key Goals

Identify key packages
Build and document guidance for business -> module usage
Document backlog of help needed by modules
Build, document and evangelize guidance, tools and processes to help maintainers

What's Next?

- Understanding the state of the ecosystem
- Support info
- Best Practices
- Develop Patterns of Engagement
- Tooling https://github.com/pkgjs

```
{ "support": {
    "target" : "LTS",
    "response": "REGULAR-7",
    "response-paid": "REGULAR-1",
    "backing": "COMPANY",
    "url": "http://mygreatmodule.org/supportinfo.html"
  }
}
```

https://github.com/nodejs/package-maintenance/blob/master/docs/drafts/PACKAGE-SUPPORT.md

Support Info

- Reducing mismatched expectations
- Closer Communication and Collaboration
- Making it easier to maintain packages
- Responsible + sustainable consumption

target: the platform versions that the package maintainer aims to support.

response: how quickly the maintainer chooses to, or is able to, respond to issues and contacts for that level of support

backing: how the project is supported

Key Attributes

JSON
- Tooling Friendly
- Still human readable
- Consistent with Package.json

General
- Tailored to JavaScript Ecosystem but applicable more broadly

Draft
- Want **your** input

https://github.com/nodejs/package-maintenance/blob/master/docs/drafts/PACKAGE-SUPPORT.md

Support Info

- Reducing mismatched expectations
- Closer Communication and Collaboration
- Making it easier to maintain packages
- Responsible + sustainable consumption

```
"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"
      }
    }
  ]
}
```

https://github.com/nodejs/package-maintenance/blob/master/docs/drafts/PACKAGE-SUPPORT.md (screenshots on pages which follow as well)

Support Info

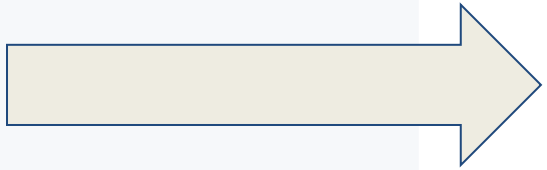
- Reducing mismatched expectations
- Closer Communication and Collaboration
- Making it easier to maintain packages
- Responsible + sustainable consumption

```
"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	Description
`none`	There is nobody backing this package
`hobby`	The single maintainer maintains the package for fun, does not get any support to continue maintenance.
`sponsored`	The single maintainer actively maintains the package but depends on sponsorship to be able to continue to maintain the package. Consider supporting this sponsorship through the funding platforms listed.
`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 project).
`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 be related to their product or service offerings.
`commercial`	The package is maintained and supported by a corporate entity as part of supporting their products.
`paid-support`	The package is maintained and supported through paid support contracts.
`freemium`	Basic version of the package it provided for free, premium version is available at a cost.
`donations`	The project can be funded by any donations.

Support Info

```
"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"
      }
    }
  ]
}
```



Reducing mismatched expectations

llaboration

ackages

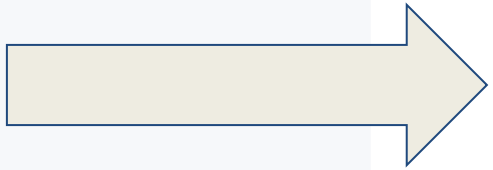
sumption

Value	Current	Active LTS	Maintenance LTS	EOL	Example	Description
xxxxxx						xxxxxx is a semver range of Node.js versions supported
abandoned						Not recommended for use. The package is 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	✔	✔	✔	✔	...,8,9,10,11,12	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		✔	✔		8,10	The package is maintained for the Node.js LTS releases (both in Active and Maintenence 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	All releases that are in active LTS
lts_active		✔			10	All releases both LTS and active. There may be more than one LTS release in active mainteance at a given point in time
lts_latest		✔			10	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	✔	✔	✔		12,10,8	Node.js versions which are not EOL
current	✔				12	The latest release from "all"

Support Info

- Reducing mismatched expectations
- Closer Communication and Collaboration
- Making it easier to maintain packages
- Responsible + sustainable consumption

```
"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"
      }
    }
  ]
}
```

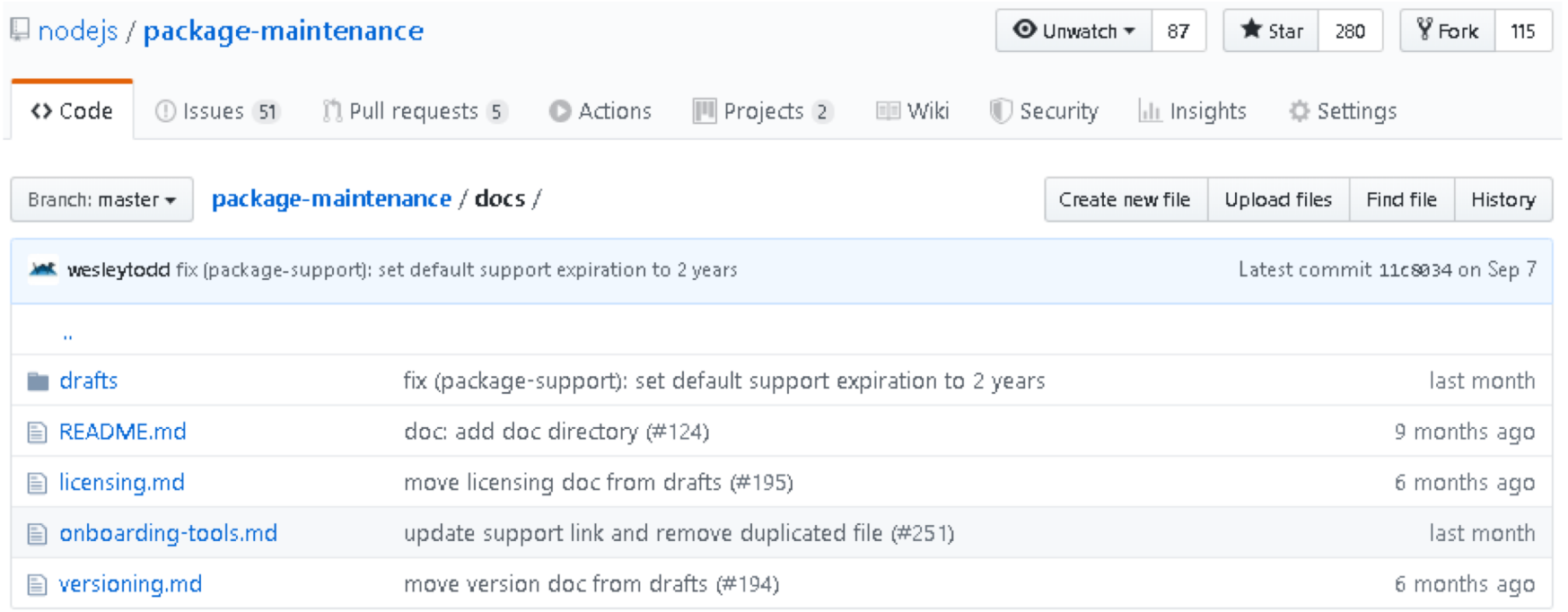


Value	Description
`none`	Don't expect a response, the package is not being actively maintained
`time-permitting`	The maintainer is interested in fixing/discussing issues, however, there should be no expectation on response times. If and when the maintainer has time they may respond.
`regular-X`	There are dedicated resources who regularly maintain the package, expected response time is X days or less for a "we read your issue" response. Further work will depend on prioritization of the issue by the maintainer team.
`24-7`	There are dedicated resources who regularly maintain the package and they are available 24/7. You can expect to be able to contact the maintainers and get an initial response with 6 hours.

Best Practices

Reducing mismatched expectations
Closer Communication and Collaboration
+ Making it easier to maintain packages
Responsible + sustainable consumption

- CI/CD
  - Testing
- Publishing
  - Support info
  - Versioning
  - Licensing
- Deprecation



Developing Patterns of Engagement

- Reducing mismatched expectations
- Closer Communication and Collaboration
- Making it easier to maintain packages
- Responsible + sustainable consumption

- Approach
  - Choose 1-2 pilot packages
  - Experiment
  - Document what “Works”
- Currently Working with Express
  - Help to triage/answering questions
  - Top ten list as identified
  - Help in moving forward key objectives.

https://github.com/nodejs/package-maintenance/issues/233
https://github.com/nodejs/package-maintenance/pull/230

Working Groups - Build/Automation

Key Goals

Support infrastructure to deliver Node.js

What's Next

Working to add IBMi to CI

https://github.com/nodejs/node/blob/master/BUILDING.md

Working Groups - Diagnostics

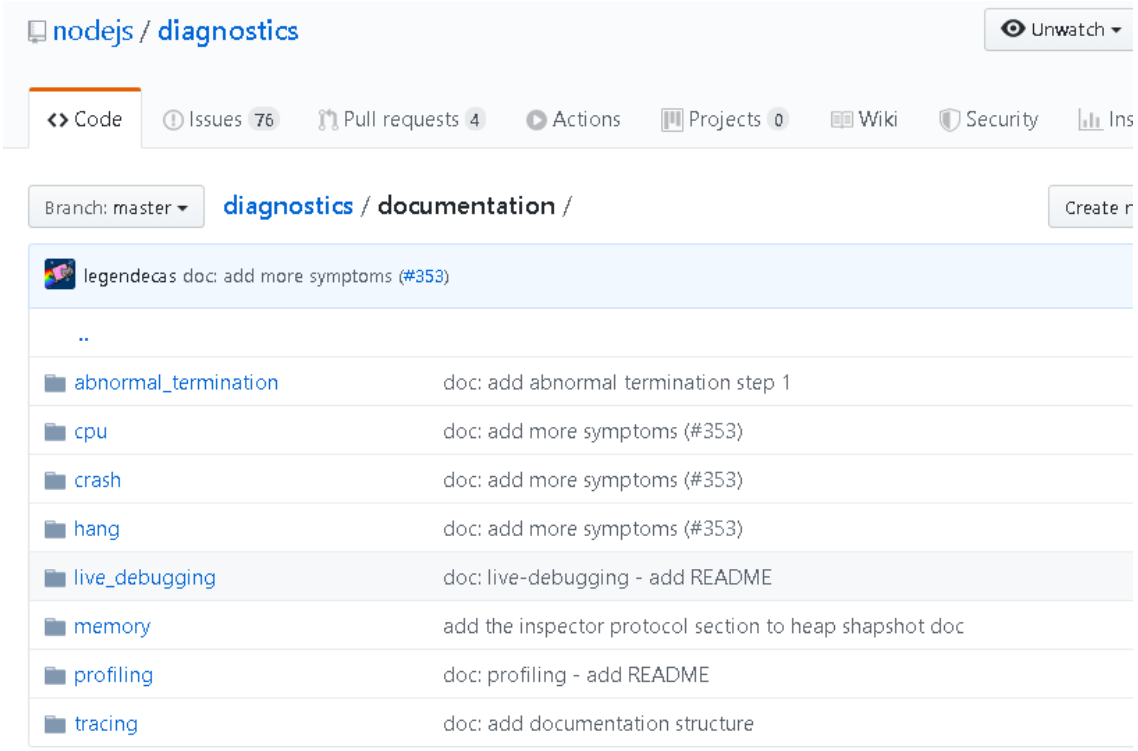
Key Goals

First class

- Tracing
- Profiling
- Heap and memory analysis
- Step debugging
- Post mortem analysis

What's Next

Best practices guides



https://github.com/nodejs/diagnostics/tree/master/documentation

Working Groups - Website

Key Goals

Great landing page and information about Node.js

What's Next?

Next generation website. Experimentation at Nodejs.dev

Strategic Initiatives

Strategic Initiatives – Modules – Experimental

- Context
  - Node.js has pre-existing module system
  - ES6 Standardized new module system
- Goals
  - Browser compatible ES6 implementation (as possible)
  - Co-existence with existing module system
- Still Experimental
- No Flag needed in 13.x

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

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

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

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

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

```
package.json can set as esm with”
“type”: “module”
```

Strategic Initiatives - N-API

- N-API is a **stable API layer for native modules**, which provides ABI **compatibility guarantees** across different Node.js versions & flavors.
  https://nodejs.org/dist/latest/docs/api/n-api.html
- N-API enables native modules to **just work** across Node.js versions without recompilations!
- A handy-dandy C++ API maintained by the Node.js organization is also available:
  https://github.com/nodejs/node-addon-api

```
#include <node_api.h>

napi_value RunCallback(napi_env env,
                       const napi_callback_info info) {
  napi_status status;
  size_t argc = 1;

  napi_value args[1];
  status = napi_get_cb_info(env, info, &argc, args,
                            nullptr, nullptr);
  napi_value cb = args[0];

  napi_value argv[1];
  status = napi_create_string_utf8(env, "hello world",
                                   NAPI_AUTO_LENGTH, argv);

  napi_value global;
  status = napi_get_global(env, &global);

  napi_value result;
  status = napi_call_function(env, global, cb, 1,
                              argv, &result);
  return nullptr;
}
```

Strategic Initiatives - N-API – What's new

- N-API 4
  - Added thread-safe function
- N-API 5
  - Added API to manage date object
  - Finalizer callback (marked as stable)
  - Optional callback in thread-safe function.
- node-addon-api (~600k downloads/week)
  - Improved documentation
  - Napi::AsyncContext
  - Napi::ThreadsafeFunction
  - Napi::AsyncProgressWorker
  - Napi::Date
- Support for
  - prebuild - https://www.npmjs.com/package/prebuild#n-api-considerations
  - cmake-js - https://www.npmjs.com/package/cmake-js#n-api-and-node-addon-api

QUIC – WIP - https://github.com/nodejs/quic

- Sequencing/backlog issues in HTTP/2
- HTTP/3 will use QUIC as transport
- UDP based transport protocol
  - UDP
    - Unreliable
    - Flow control issues
- Adds in
  - Error Handling
  - Acknowledgements
  - Flow Control
  - Sequencing
  - Encryption (TLS 1.3 mandatory)
  - Bidirectional/Unidirectional streams
- Connections are not tied to routing (can switch Wifi without dropping connections)

https://en.wikipedia.org/wiki/QUIC

https://github.com/nodejs/quic

So What's Next?

It's what YOU make it!

Meet us in GitHub

Summary and Questions