
Node.js What's Next?

Michael Dawson
IBM Community Lead for Node.js

Agenda Key: 27AH

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?)
- High level predictions
- Releases
- Working Groups/Teams
- Strategic Initiatives
- Foundation 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



Go Fishing, How to track What's Next

- Releases

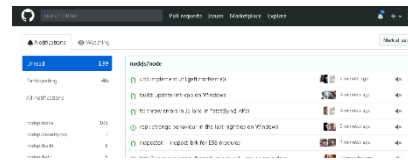


<https://nodejs.org>

- Twitter

<https://twitter.com/gibfahn/lists/node-js-news>

- 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>

- Foundation Initiatives



<https://foundation.nodejs.org>
<https://openjsf.org/>

High Level Predictions

- Greater cross project collaboration
- Increased Standards involvement (TC39 and others)
- Emphasis on Diagnostics
 - Diagnostic Report
 - Heapdump, Memory Sampling Profiler
 - CLS API?
- Progress on helping Package Maintainers
- Continued focus on user feedback
- Growth of use of ES6 Modules
- Exploration of Worker use cases
- Promisification of Core

Releases - Process

Release Types

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

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

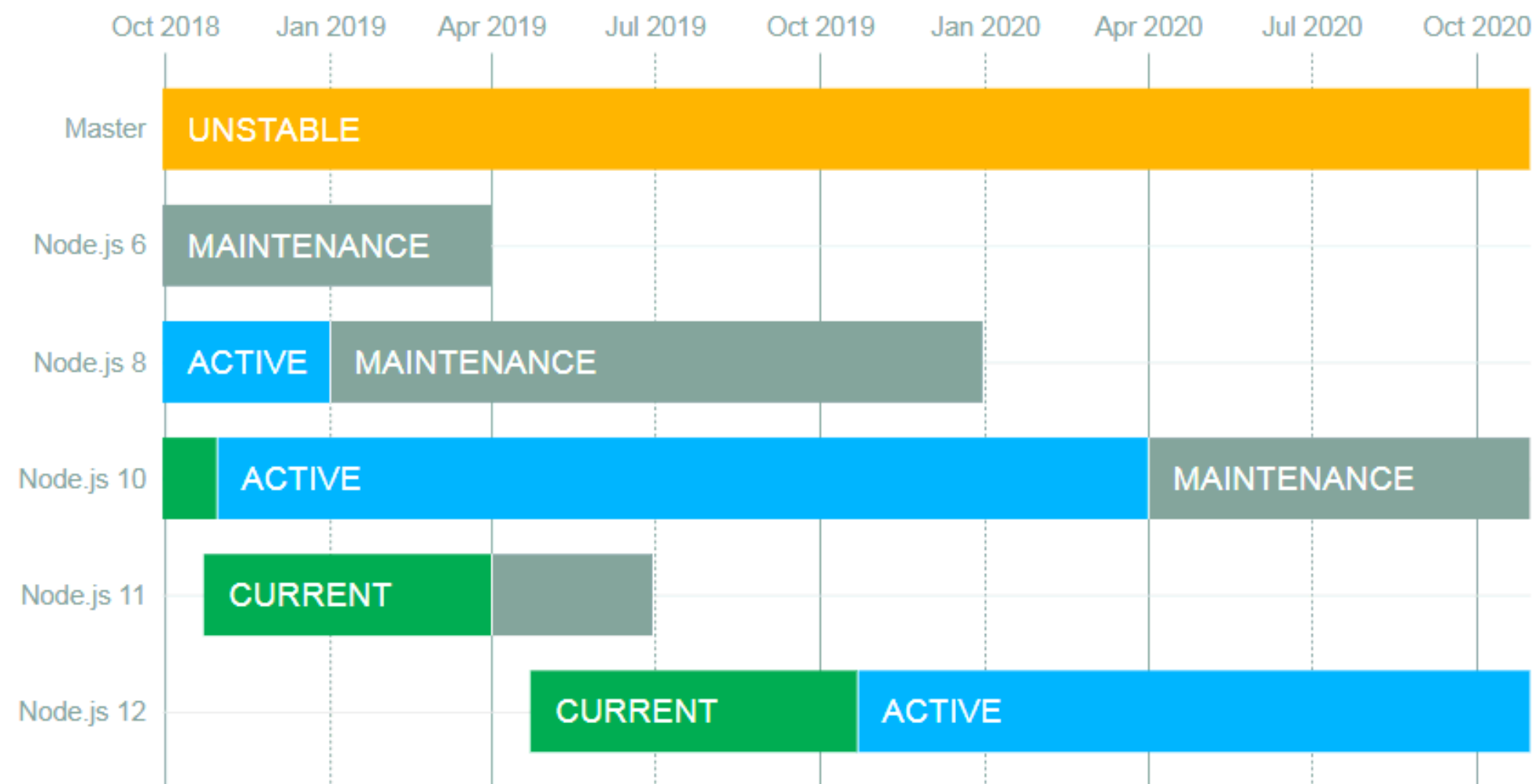
<https://github.com/nodejs/node/blob/master/CHANGELOG.md>

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 2019



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

Working Groups and Teams

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?

- Engage with pilot packages
- First round of baseline practices
 - support levels
 - testing
 - licensing
 - versioning
- Guidance for businesses
- Fist backlog entries

```
{ "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/Baseline%20practice%20-%20Document%20support%20levels.md>

Strategic Initiatives - User Feedback

Key Goals

Increase connection and discourse with our Users

What's Next

Surveys

Promises

Website

TypeScript

User Feedback at meetups

Enterprise subgroup

Working Groups - Build/Automation

Key Goals

Support infrastructure to deliver Node.js

What's Next

More automation tools/scripts

Changes to supported OS levels

- Incremental platform test coverage

- Dropping coverage for older OS's

- Addition of newer OS's

Requirement for higher gcc levels

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

Working Groups - Release

Key Goals

Predicable, regular Release

Stable releases

What's Next

No major changes planned to release process

More automation

Increase/improve safety net

Easier to find/consume releases

More visible release schedule

Working Groups - Benchmarking

Key Goals

Avoid Regressions

Track and Evangelize gains between releases

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

How ?

Define Use Cases

Identify/Build Benchmarks

Run/Capture results

What's Next

New workloads, promises?



<https://benchmarking.nodejs.org/>

Working Groups - Security-wg

Key Goals

- Define and Document security policies
- Help implement those policies

What's Next?

- Updates to Vulnerability database format

Working Groups - Diagnostics

Key Goals

First class

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

What's Next

Best practices guides

Better testing for tools (Tiers)

Hardening of Async Hooks

Async Context formalization

Greater integration (node-report (Done – Yay), heapdump, memory samplers, llnode metadata maintenance)

Leveraging new trace engine

Working Groups - Website

Key Goals

Great landing page and information about Node.js

What's Next?

Next generation website. Experimentation at [Nodejs.dev](https://nodejs.dev)

Strategic Initiatives

Strategic Initiatives - Http2

- 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
 - <https://en.wikipedia.org/wiki/QUIC>
- <https://github.com/nodejs/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);
```

Strategic Initiatives - Modules

- 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
- Experimental version in Node.js master

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

`--experimental-modules`

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

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

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

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

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 - Async Hooks

- Context across Asynchronous calls

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

```
function init(asyncId, type, triggerAsyncId, resource) { }  
function before(asyncId) { }  
function after(asyncId) { }  
function destroy(asyncId) { }  
function promiseResolve(asyncId) { }
```

```
const asyncHook = async_hooks.createHook({ init, before, after, destroy, promiseResolve });
```

```
asyncHook.enable();  
asyncHook.disable();
```

FSEVENTWRAP, FSREQWRAP, GETADDRINFOREQWRAP, GETNAMEINFOREQWRAP, HTTPPARSER,
JSSTREAM, PIPECONNECTWRAP, PIPEWRAP, PROCESSWRAP, QUERYWRAP, SHUTDOWNWRAP,
SIGNALWRAP, STATWATCHER, TCPCONNECTWRAP, TCPSERVER, TCPWRAP, TIMERWRAP, TTYWRAP,
UDPSENDWRAP, UDPWRAP, WRITWRAP, ZLIB, SSLCONNECTION, PBKDF2REQUEST,
RANDOMBYTESREQUEST, TLSWRAP, Timeout, Immediate, TickObject

Strategic Initiatives - Core Promise APIs

- Support functions added in **Node.js v8.2.0**

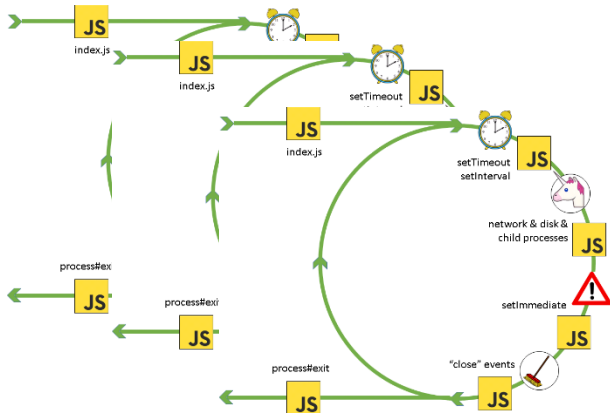
- `util.promisify()`
 - Accepts a “nodeback”-style function (`error, ...args`), returns a `Promise`
 - Not context-aware
- `util.callbackify()`
 - Accepts a `Promise`-returning function, returns a “nodeback”-style function
- **Experimental** `Promise`-based API for `fs` module added in **v10.0.0**
- `fs.promises`
- See docs:
https://nodejs.org/api/fs.html#fs_fs_promises_api

```
const fs = require('fs').promises;

async function copy(src, dest) {
  try {
    await fs.copyFile(src, dest);
    console.log(`copied ${src} to ${dest}`);
  } catch (err) {
    console.log(`copy failed: ${err}`);
  }
}
```

Strategic Initiatives - Workers

- 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
 - Experimental (but no flag needed in Node.js v11.7+)



```
// 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
```


Strategic Initiatives – Diagnostic Reports

- 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
```

Foundation Initiatives

Foundation Initiatives

Node.js + JS Foundation Merged



Foundation

- A common home for JavaScript projects
- Removes confusion over “which” foundation to join
- Stronger and better funded

Projects

- Stronger/better home for your code
- Better Collaboration
- Better support for the projects you depend on
- Common Foundation to engage for key concerns

Summary and Questions

Don't Forget Your Session Survey!

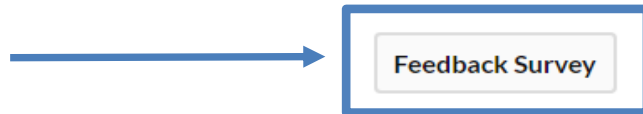
Sign in to the Online Session Guide

(www.common.org/sessions)

Go to your personal schedule

Click on the session that you attended

Click on the *Feedback Survey* button located above the abstract.



Come to this session to learn about the DB2 for IBM i enhancements delivered in 2016. This session will include reasons why you should upgrade to the latest IBM i release.
This is session 610533

Completing session surveys helps us plan future programming and provides feedback used in speaker awards. Thank you for your participation.