This.dot State of JS Frameworks Update

Michael Dawson @ IBM

13 Aug 2019

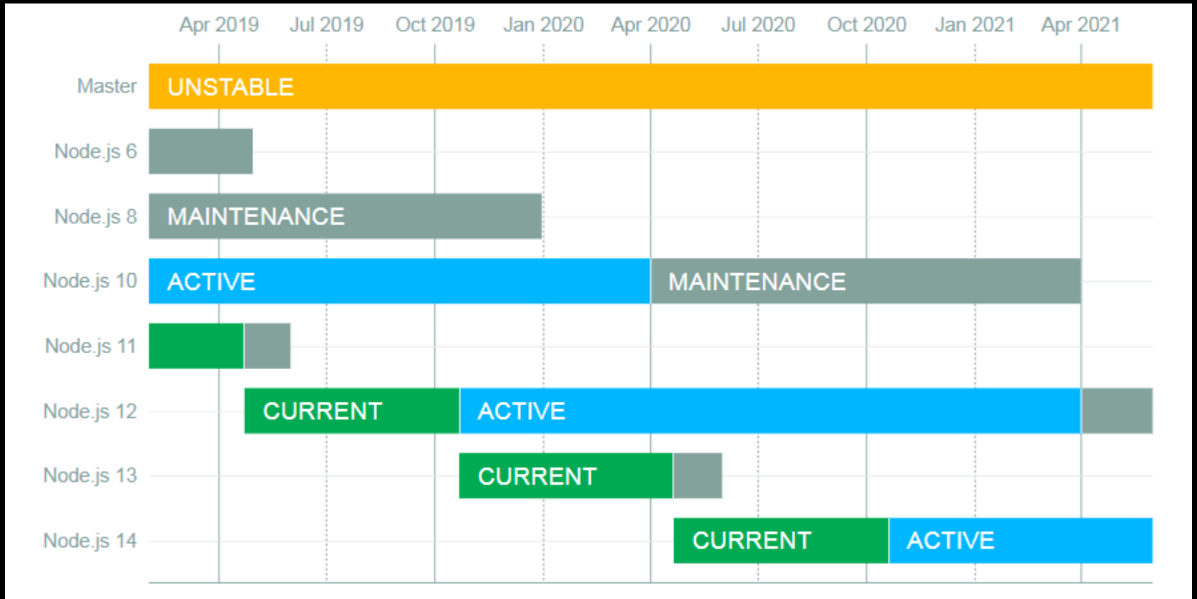
About Michael Dawson
IBM Community Lead for Node.js

- Active Node.js community member
  - Collaborator
  - Node.js Technical Steering Committee TSC Chair
  - Community Committee member
  - Involved in many Working group(s)/Teams
  - Node.js OpenJS Board Representative

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

Releases

- Upcoming LTS
  - Node.js 12.X – starting Oct 2019
- Going EOL
  - Node.js 8.X– Dec 2019 (early)
- Ongoing LTS Releases
  - Node.js 10.X – ends April 2021
- Current
  - Now 12.x,
  - 13.x in October
- Some discussion around swapping Active/Maintenance times



https://github.com/nodejs/Release

Key New Features in 12.X

- V8 Update to - 7.5 (likely to be 7.6 by LTS)
  - Async stack traces
  - Perf, mismatched arguments, await, javascript parsing
- Diagnostic Improvements
  - Diagnostic Report (Experimental but no flag)
  - Heap Dumps
- Performance Improvements
  - Faster Startup
  - Worker threads (Experimental but no flag)
  - Auto heap limit configuration
- Security
  - TLS 1.3
- Modules
  - ES6 Module support
  - Continued progress on N-API
- General
  - llhttp http parser
  - New Compiler and platform minimums

https://medium.com/@nodejs/introducing-node-js-12-76c41a1b3f3f

Features In Progress

- Support For Python 3
- New Streams https://github.com/Fishrock123/bob
- Quic https://github.com/nodejs/quic
- Workers https://nodejs.org/api/worker_threads.html
  - Experimental in Core
- Node Report https://www.npmjs.com/package/node-report
  - Experimental in core
- ES6 Modules https://github.com/nodejs/modules
- Core Promise APIs https://nodejs.org/api/fs.html#fs_fs_promises_api
  - Experimental in core (ex fs)
  - Validate approach
- Async Hooks https://github.com/nodejs/diagnostics/tree/master/async-context/slide-decks
  - Experimental in core
  - Formalization of terminology

Workers

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

- Based on Web Workers (https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API/Using_web_workers)
- No flag needed in 11.7 or later but still Experimental
- Each thread is separate JavaScript environment
- Data can be exchanged through messaging:
  - Object cloning
    - structured clone algorithm
  - Handoff (transferList/ArrayBuffer)
  - Memory sharing – SharedArrayBuffer/Atomics

    https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/SharedArrayBuffer
    https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Atomics
- Cannot transfer handles
- Re-use/Pooling is recommended

Workers

```
const workers = require('worker_threads');

if (workers.isMainThread) {
  const worker1 = new workers.Worker(__filename, { data1: '1234' });
  const worker2 = new workers.Worker(__filename, { data2: '4567' });

  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);
  });
}
```

```
parent:running
parent:running
worker:wakeup:2
parent:wakeup
```

Report

- Initially an NPM module
- Now integrated into core in master (Experimental)

https://www.npmjs.com/package/node-report

https://nodejs.org/docs/latest/api/report.html

```
--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
```

```
process.report.getReport([err])
process.report.setDiagnosticReportOptions([options]);
process.report.triggerReport([filename][, err])
```

Report

report.20190129.200907.114.001.jsonn

```
node –experimental-report –diagnostic-report-uncaught-exception test.js
```

```
throw new Error('test')
```

But its not only about Features

- Teams, Working Groups and Strategic Initiatives
  - Package Maintenance (https://github.com/nodejs/package-maintenance)

  - Tooling
  - Website re-design
  - Mentoring
  - User Feedback
  - Internationalization
  - Security
  - Diagnostics
  - Build
  - Automation
  - Benchmarking
  - Release
  - Application Showcase

Building the Safety Net
Growing Involvement
Leveraging better information
Supporting our end users
Makings things easier

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

Package Maintenance

- Repository for discussion on how to help ensure baseline maintenance and ability to safely use key packages in the ecosystem with current Node.js versions. You can find more about this initiative in the article: Call to Action: Accelerating Node.js Growth

- Support Meta Data - https://github.com/nodejs/package-maintenance/pull/220
  - **target:** the level of support that the package maintainer aims to provide
  - **response**: how quickly the maintainer chooses to, or is able to, respond to issues
  - **response-paid**: how quickly the maintainer will respond to issues if paid
  - **backing**: how the project can be supported

Node is now part of OpenJS Foundation

- CPC now up and running
  - https://github.com/openjs-foundation/cross-project-council
  - Members
    - Voting
    - Regular
    - Observers
  - Still working out overall governance
- Project progression defined
- WG/teams starting to form
  - Standards - https://github.com/openjs-foundation/standards

So What’s Next?

It’s what YOU make it!

Meet us in GitHub

Copyright and Trademarks