The Module Ecosystem
The Stress of Success

NodeConf.eu 2019

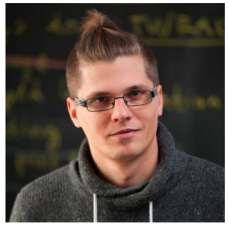
Dominykas Blyžė (NearForm)

Michael Dawson (IBM)

https://gist.github.com/dominykas/13eb027ca5343ebbdcd32b690c0dfaec

About Dominykas

- Developer at NearForm
- Long time lurker
- Trying to get through the day,
  which sometimes means working on OSS



- Twitter: @eDominykas
- Github: @dominykas

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
  - Node.js OpenJS Foundation Board Member
  - Working group(s) member/leadership

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

Agenda

- The problems
- How can we make things Better?
- What we’re doing
- Call to Action

The problems

- Explosive Growth -> Dependency on Key modules

The problems

- Explosive Growth -> Dependency on Key modules
- Desire for basic maintenance

The problems

- Explosive Growth -> Dependency on Key modules
- Desire for basic maintenance
- Maintainers struggling to keep up

The problems

- Explosive Growth -> Dependency on Key modules
- Desire for basic maintenance
- Maintainers struggling to keep up
- Increasing worry about dependency tree

The problems

- Explosive Growth -> Dependency on Key modules
- Desire for basic maintenance
- Maintainers struggling to keep up
- Increasing worry about dependency tree
- Lack of communication channels

To read more:

https://medium.com/@nodejs/call-to-action-accelerating-node-js-growth-e4862bee2919

https://blog.acolyer.org/2019/09/30/small-world-with-high-risks/

Differing Needs

- Want support to do it themselves

Differing Needs

- Want support to do it themselves

- Want to grow contribution

Differing Needs

- Want support to do it themselves
- Want to grow contribution
- Want to move on

Making things better

Making things better

- Reducing mismatched expectations

Making things better

- Reducing mismatched expectations
- Closer Communication and Collaboration
  - Between consumers and maintainers
  - Between maintainers
  - And everybody else...

Making things better

- Reducing mismatched expectations
- Closer Communication and Collaboration
  - Between consumers and maintainers
  - Between maintainers
  - And everybody else...
- Making it easier to maintain packages

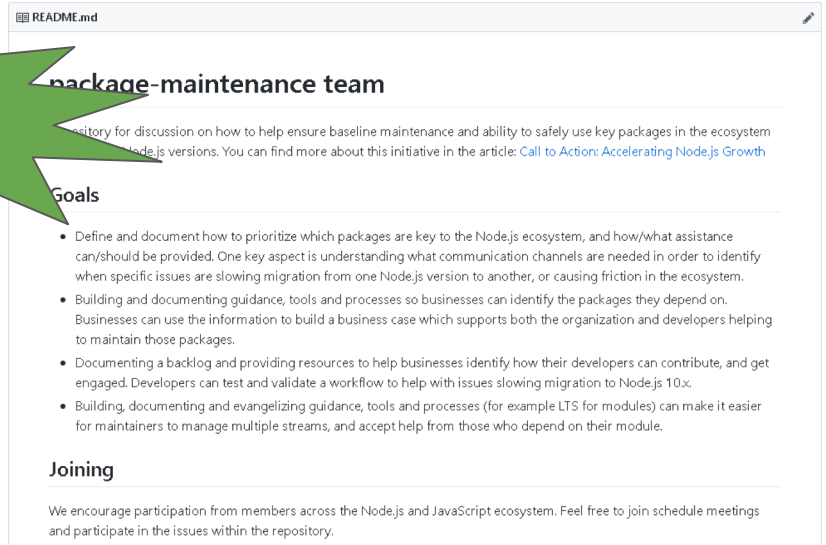
Making things better

- Reducing mismatched expectations
- Closer Communication and Collaboration
  - Between consumers and maintainers
  - Between maintainers
  - And everybody else...
- Making it easier to maintain packages
- Promoting responsible + sustainable consumption

The Node.js package maintenance team

- History
  - Module LTS
  - Event Stream
- Representation
  - Consumers
  - Package Maintainers
  - Node.js collaborators
  - Npm
  - **You?**
- A place to work together
  - Share processes, practices, tooling, manage ecosystem backlog

Launched ~ 1 year ago



README.md

package-maintenance team

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

Goals

- Define and document how to prioritize which packages are key to the Node.js ecosystem, and how/what assistance can/should be provided. One key aspect is understanding what communication channels are needed in order to identify when specific issues are slowing migration from one Node.js version to another, or causing friction in the ecosystem.
- Building and documenting guidance, tools and processes so businesses can identify the packages they depend on. Businesses can use the information to build a business case which supports both the organization and developers helping to maintain those packages.
- Documenting a backlog and providing resources to help businesses identify how their developers can contribute, and get engaged. Developers can test and validate a workflow to help with issues slowing migration to Node.js 10.x.
- Building, documenting and evangelizing guidance, tools and processes (for example LTS for modules) can make it easier for maintainers to manage multiple streams, and accept help from those who depend on their module.

Joining

We encourage participation from members across the Node.js and JavaScript ecosystem. Feel free to join schedule meetings and participate in the issues within the repository.

https://github.com/nodejs/package-maintenance

The Node.js package maintenance team

- What we're doing today
  - Understanding the state of the ecosystem
  - Support info
  - Best Practices
  - Develop Patterns of Engagement
  - Tooling

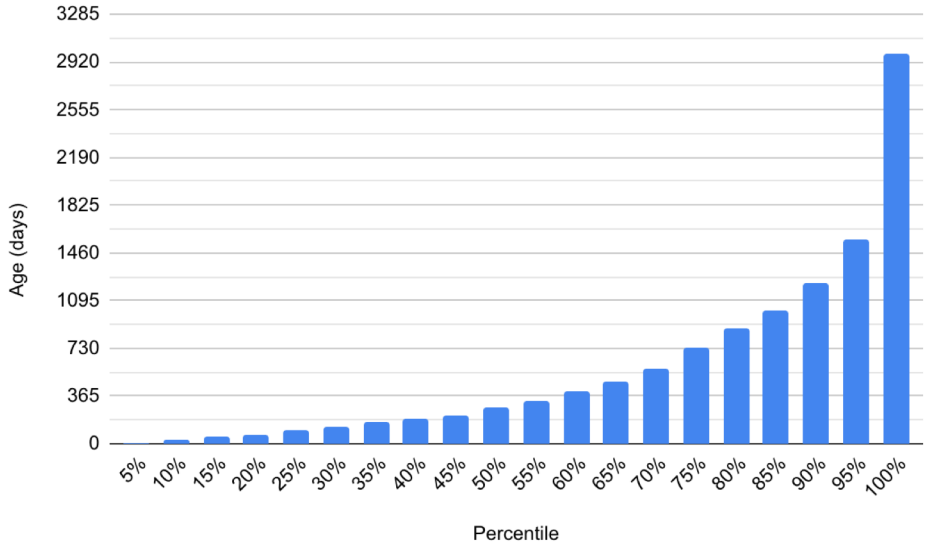
State of the Ecosystem: Surveys

- What problems are maintainers facing?
- What are the most time consuming tasks?
- Understanding the impact of dependencies

https://github.com/nodejs/package-maintenance/tree/master/pilots

State of the Ecosystem: Release age

- **~60%** of packages had a release **in the last year**
- **~40%** of packages had a release **in the last 6 months**
- **150 had commits** since v10 came out, **but did not test in v10**

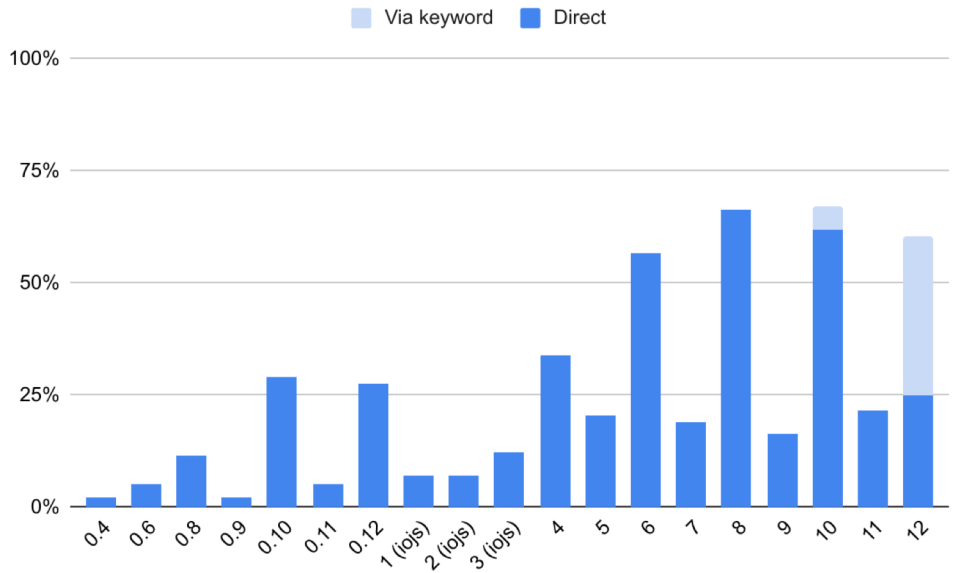


Fun fact: longest time since last release in top 1000 - substack's jsonify@0.0.0, published in Aug/2011

https://docs.google.com/spreadsheets/d/1lZDNYsLntwD2q9XaTw-XLuG0VpHH6cOV-Uoa7Y1aTSM/edit#gid=1745448509

State of the Ecosystem: Node versions in CI

- **877** packages point to repos with **.travis.yml**
- Just over **2/3** were testing in **active LTS**
- **70 packages** added v10 to the matrix since April 2019



Fun fact: 1 package wanted to test in 13 before it is out

https://medium.com/@nodejs/choosing-the-node-js-versions-for-your-ci-tests-hint-use-lts-89b67f68d7ca

State of the Ecosystem: Outdated dependencies

- 400 out of top 1000 have no production dependencies
- **20% of dependencies outdated** (376 of 1968)
  - 212 packages have at least one outdated dependency
- Outdated dev dependencies - up to 47%
  - Information from npm, not git - could be updated, but not published
  - 713 packages have at least one outdated dev dependency

Fun fact: installing top 1000 packages results in 380MiB of node_modules

https://docs.google.com/spreadsheets/d/1ciqXf9siAbI_re-laF4KoEYYN3ZujRJ-QJElEURld-I/edit#gid=0

State of the Ecosystem: Known vulnerabilities

- 7 packages have deprecation warnings
- 0 vulnerabilities reported by npm audit 🎉
- 0 vulnerabilities reported by snyk 🎉

Fun fact: cloning the repos of top 1000 packages results in 4.8GiB of data

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

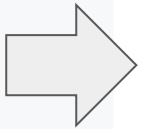
```
"support": {
  "versions": [
    {
      "version": "*",
      "target": {
        "node": "none"
      },
      "response": {
        "type": "time-permitting",
        "paid": false,
        "contact": {
          "name": "Volunteers",
          "url": "https://github.com/myproject"
        }
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

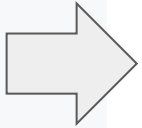


Value	Description
`none`	There is nobody backing this package
`hobby`	The single maintainer maintains the package for fun, does not get any support to continue maintenance.
`sponsored`	The single maintainer actively maintains the package but depends on sponsorship to be able to continue to maintain the 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

- 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`	Th… or…
`commercial`	Th…
`paid-support`	Th…
`freemium`	Ba…
`donations`	Th…

```
{
  "sponsored": [
    "https://opencollective.com/my-account",
    "https://www.patreon.com/my-account",
    "https://tidelift.com/subscription/pkg/my-package"
  ]
}
```

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"
      }
    }
  ]
}
```

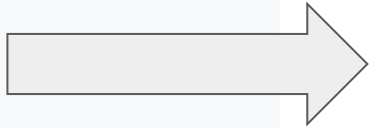
Reducing mismatched expectations
d Collaboration
in packages
e consumption

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"
      }
    }
  ]
}
```

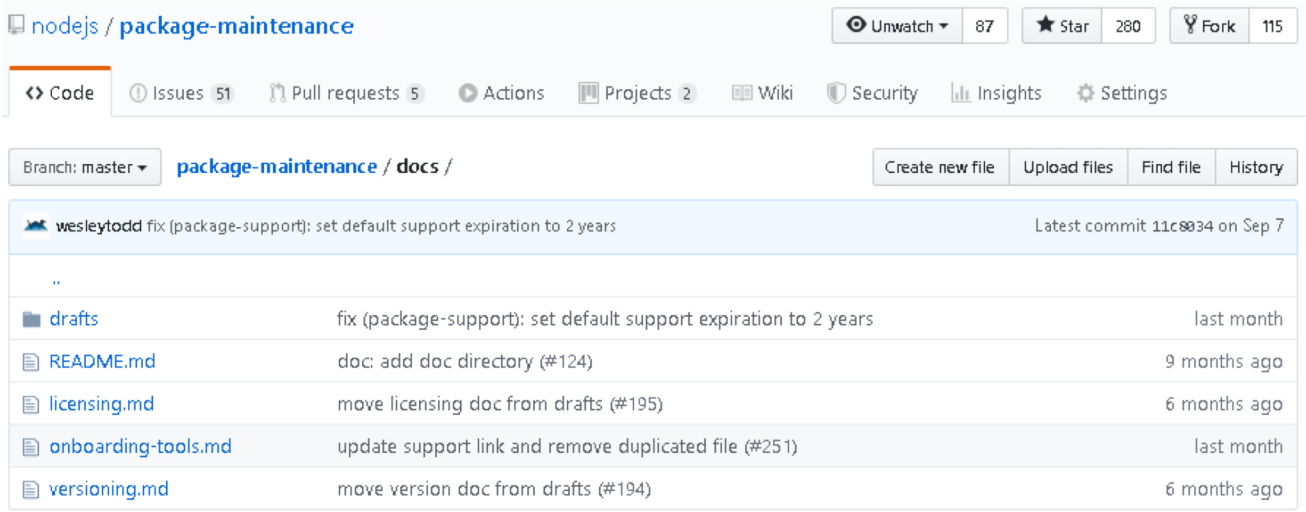


Value	Description
`none`	Don't expect a response, the package is not being actively maintained
`time-permitting`	The maintainer is interested in fixing/discussing issues, however, there should be no expectation on response times. If and when the maintainer has time they may respond.
`regular-X`	There 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

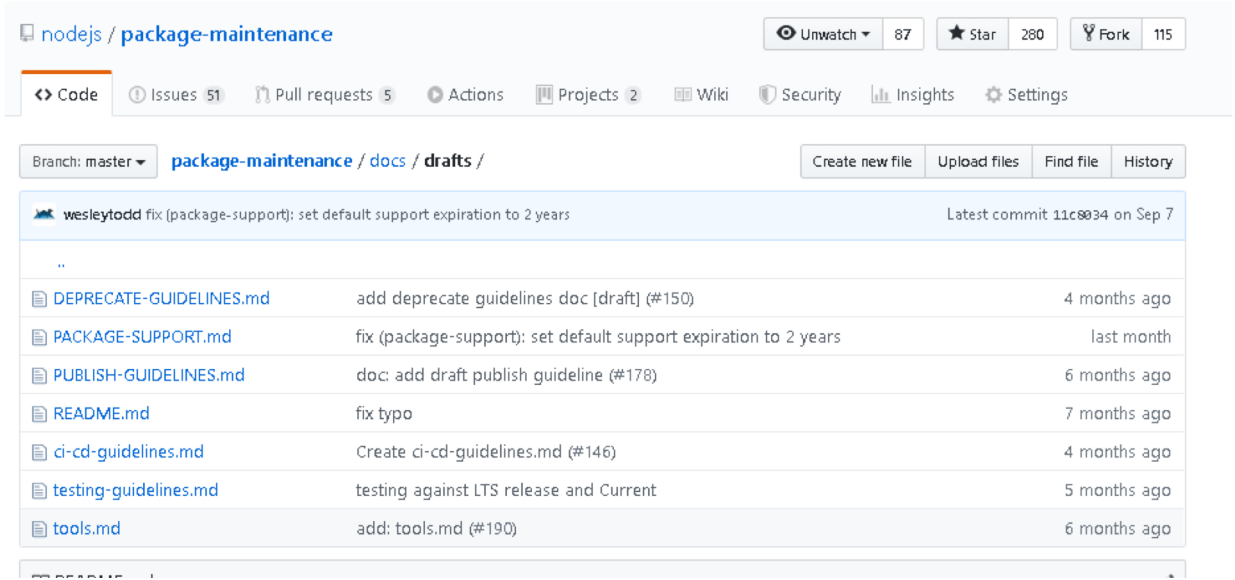
- CI/CD
  - Testing
- Publishing
  - Support info
  - Versioning
  - Licensing
- Deprecation



Best Practices

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

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



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”

Developing Patterns of Engagement

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

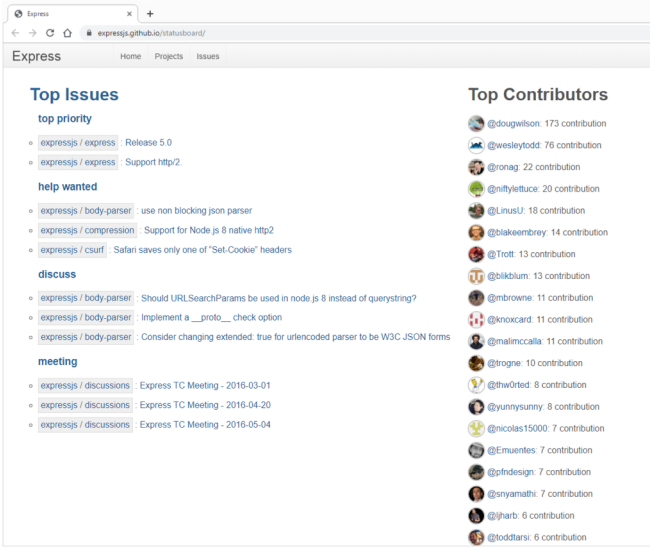
- Approach
  - Choose 1-2 pilot packages
  - Experiment
  - Document what “Works”
- Currently Working with Express
  - Help to triage/answering questions
  - Top ten list as identified
  - Help in moving forward key objectives.

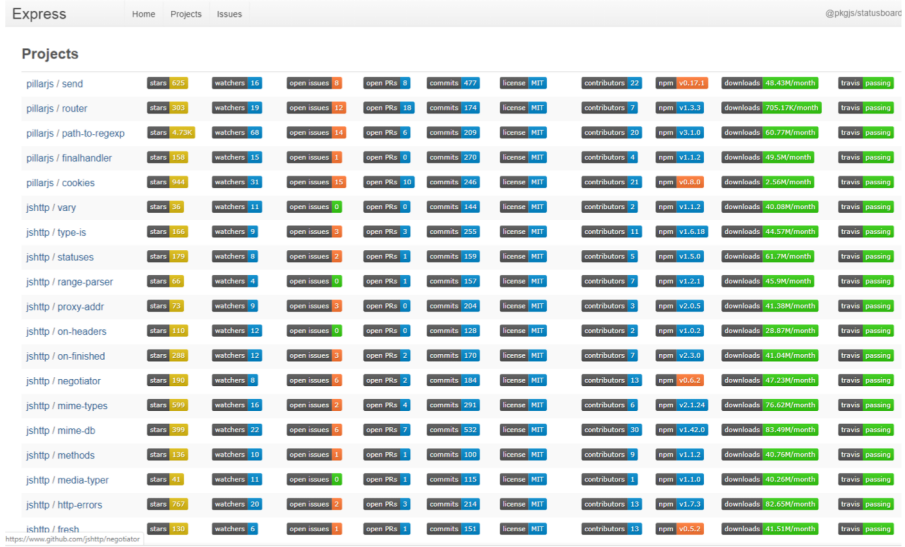
https://github.com/nodejs/package-maintenance/issues/233
https://github.com/nodejs/package-maintenance/pull/230

Developing Patterns of Engagement

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

- @wesleytodd - status board



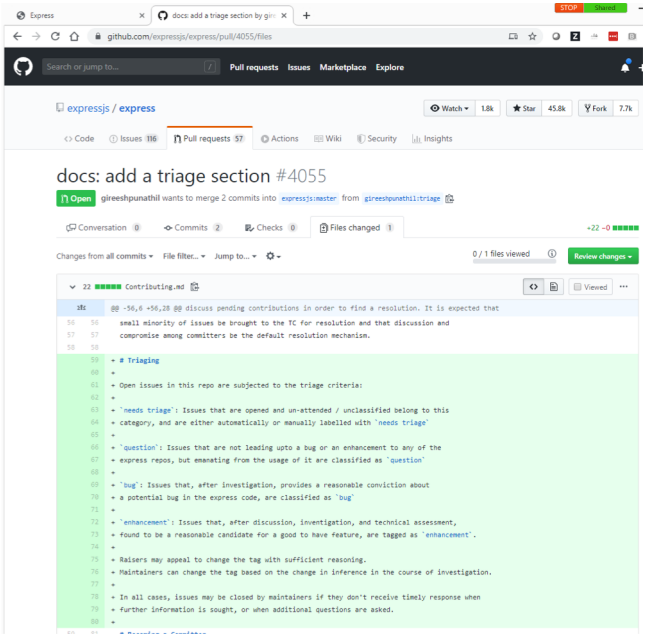


https://expressjs.github.io/statusboard

Developing Patterns of Engagement

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

- @gireeshpunathil/@wesleytodd - Triage
- https://github.com/expressjs/express/pull/4055



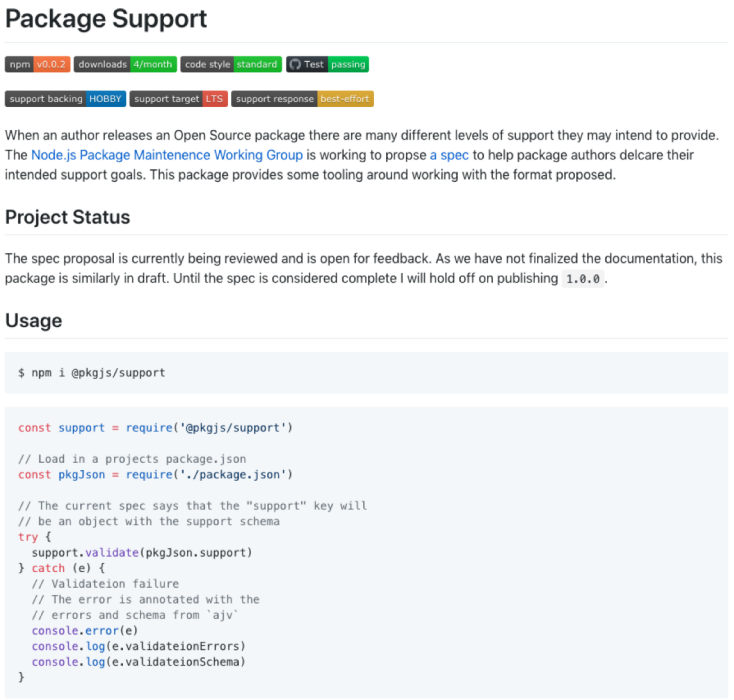
Tooling

- Built and in-progress:
  - @pkgjs/statusboard
  - @pkgjs/nv
  - @pkgjs/support
- Discussions on-going:
  - Remote 2FA, incl. for teams
- Would like to solve:
  - LTS
  - CITGM

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

Tooling: @pkgjs/support

Validates package support JSON file



Package Support

npm v0.0.2 | downloads 4/month | code style standard | Test passing

support backing HOBBY | support target LTS | support response best-effort

When an author releases an Open Source package there are many different levels of support they may intend to provide. The Node.js Package Maintenence Working Group is working to propse a spec to help package authors delcare their intended support goals. This package provides some tooling around working with the format proposed.

Project Status

The spec proposal is currently being reviewed and is open for feedback. As we have not finalized the documentation, this package is similarly in draft. Until the spec is considered complete I will hold off on publishing `1.0.0`.

Usage

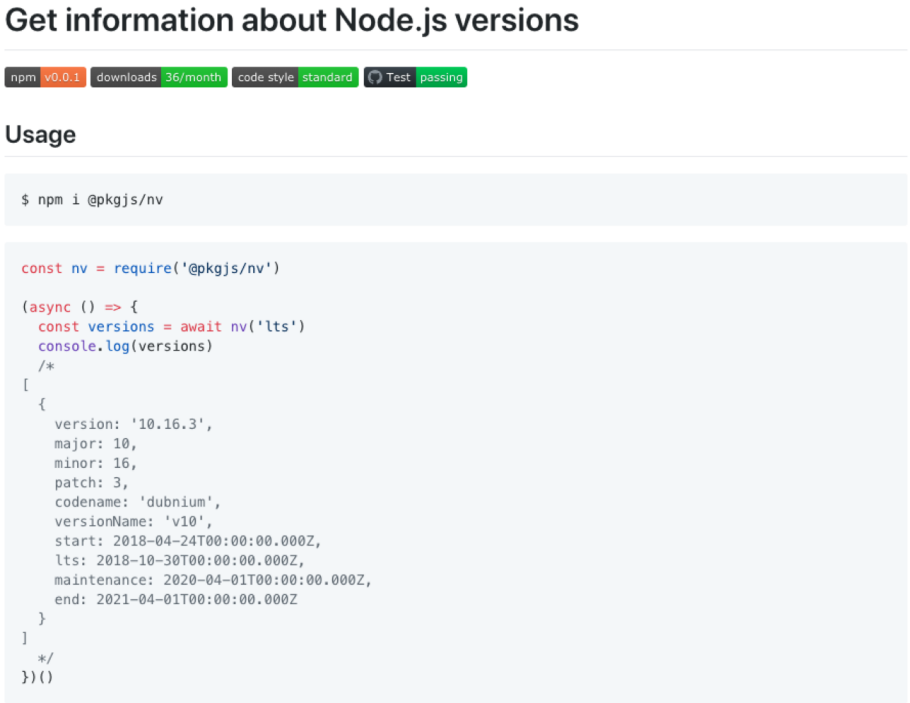
```
$ npm i @pkgjs/support
```

```
const support = require('@pkgjs/support')

// Load in a projects package.json
const pkgJson = require('./package.json')

// The current spec says that the "support" key will
// be an object with the support schema
try {
  support.validate(pkgJson.support)
} catch (e) {
  // Validateion failure
  // The error is annotated with the
  // errors and schema from `ajv`
  console.error(e)
  console.log(e.validateionErrors)
  console.log(e.validateionSchema)
}
```

Tooling: @pkgjs/nv

Resolve keywords defined in the package support draft (`lts`, `lts_active`, etc) into a list of Node.js versions

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



Get information about Node.js versions

npm v0.0.1 | downloads 36/month | code style standard | Test passing

Usage

```
$ npm i @pkgjs/nv
```

```
const nv = require('@pkgjs/nv')

(async () => {
  const versions = await nv('lts')
  console.log(versions)
  /*
[
  {
    version: '10.16.3',
    major: 10,
    minor: 16,
    patch: 3,
    codename: 'dubnium',
    versionName: 'v10',
    start: 2018-04-24T00:00:00.000Z,
    lts: 2018-10-30T00:00:00.000Z,
    maintenance: 2020-04-01T00:00:00.000Z,
    end: 2021-04-01T00:00:00.000Z
  }
]
  */
})()
```

Tooling: @pkgjs/statusboard

Get an overview of a large organization with multiple repositories

expressjs / vhost	stars 600	watchers 22	open issues 3	open PRs 3	commits 92	license MIT	contributors 2	npm v3.0.2	downloads 593.57K/month	travis passing
expressjs / timeout	stars 237	watchers 14	open issues 3	open PRs 2	commits 140	license MIT	contributors 8	npm v1.9.0	downloads 664.23K/month	travis passing
expressjs / session	stars 4.51K	watchers 115	open issues 47	open PRs 22	commits 581	license MIT	contributors 30	npm v1.17.0	downloads 2.85M/month	travis passing
expressjs / serve-static	stars 1.1K	watchers 25	open issues 8	open PRs 3	commits 307	license MIT	contributors 10	npm v1.14.1	downloads 45.68M/month	travis passing
expressjs / serve-index	stars 292	watchers 15	open issues 8	open PRs 15	commits 266	license MIT	contributors 13	npm v1.9.1	downloads 20.59M/month	travis passing
expressjs / response-time	stars 386	watchers 21	open issues 2	open PRs 2	commits 97	license MIT	contributors 3	npm v2.3.2	downloads 809.2K/month	travis passing
expressjs / multer	stars 7.03K	watchers 121	open issues 190	open PRs 36	commits 266	license MIT	contributors 30	npm v1.4.2	downloads 2.75M/month	travis passing
expressjs / morgan	stars 5.09K	watchers 84	open issues 8	open PRs 9	commits 301	license MIT	contributors 17	npm v1.9.1	downloads 8.44M/month	travis passing
expressjs / flash	stars 76	watchers 13	open issues 8	open PRs 3	commits 9	license MIT	contributors 2	npm v1.1.0	downloads 8.82K/month	travis error
expressjs / express	stars 45.79K	watchers 1.82K	open issues 116	open PRs 57	commits 5.56K	license MIT	contributors 30	npm v4.17.1	downloads 44.38M/month	travis passing
expressjs / errorhandler	stars 341	watchers 15	open issues 0	open PRs 0	commits 216	license MIT	contributors 6	npm v1.5.1	downloads 2.65M/month	travis passing
expressjs / discussions	stars 25	watchers 27	open issues 62	open PRs 0	commits 2	license no license	contributors 1			
expressjs / csurf	stars 1.64K	watchers 38	open issues 16	open PRs 6	commits 272	license MIT	contributors 18	npm v1.10.0	downloads 1.12M/month	travis passing
expressjs / cors	stars 4.18K	watchers 98	open issues 8	open PRs 2	commits 261	license MIT	contributors 30	npm v2.8.5	downloads 10.09M/month	travis passing
expressjs / compression	stars 2.03K	watchers 53	open issues 13	open PRs 15	commits 332	license MIT	contributors 12	npm v1.7.4	downloads 26.36M/month	travis passing
expressjs / body-parser	stars 4.12K	watchers 98	open issues 18	open PRs 10	commits 496	license MIT	contributors 19	npm v1.19.0	downloads 46.53M/month	travis passing

Tooling: publishing with 2FA from CI

- 0.6% of all packages and 6.89% of all maintainers have 2FA on
  - https://github.com/nodejs/package-maintenance/issues/244#issuecomment-534814168
- npm will disclose if maintainers use 2FA
  - https://blog.npmjs.org/post/188234999089/new-security-insights-api-sneak-peek
- Publishing from dev machine is tricky
  - Must ensure shared configuration for builds
  - Potential for human error in build steps
- Publishing from CI is risky
  - No built-in way to provide the second factor
- https://github.com/nodejs/package-maintenance/issues/244

Tooling: publishing with 2FA from CI - options

- Release manager
  - Cons: complexity, compatibility, effort to build, maintenance
- Remote OTP entry (mobile)
  - POC: https://github.com/nearform/optic
  - Cons: dependency on Firebase, needs a server
- Remote OTP entry (chatbot)
  - POC: ask https://twitter.com/MarshallOfSound
  - Cons: dependency on Slack, needs a server
- Remote OTP entry (SaaS)
  - Cons: none publically available and free; trust
- Use GitHub releases / alternative registries for staging
  - Cons: requires manual action and setup to complete publishing

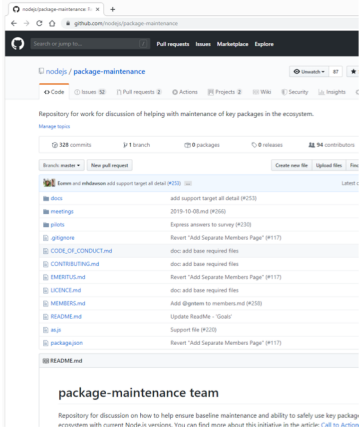
Tooling: future

- Long Term Support
  - Guidelines
  - Release automation
  - https://github.com/nodejs/package-maintenance/issues/172
- Canary In The Gold Mine
  - Validate the impact of package changes on the rest of the ecosystem
  - https://github.com/nodejs/package-maintenance/issues/84
  - https://github.com/nodejs/package-maintenance/issues/179
  - Module Insights from IBM: https://modules.cloudnativejs.io/

Call To Action

- Help us figure all this out
  - Every 2 weeks (9AM EST, 3PM EST)
  - Github
- Validate/Comment on best practices
- Let us know what works for you

github.com/nodejs/package-maintenance



nodejs.org/calendar

https://gist.github.com/dominykas/13eb027ca5343ebbdcd32b690c0dfaec

Copyright and Trademarks