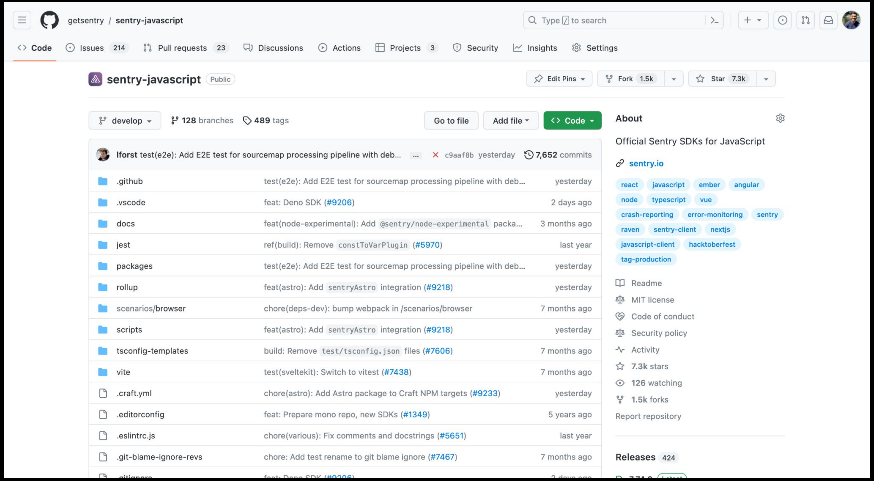
Publishing JavaScript Libraries Made Easy

Abhijeet Prasad
Software Engineer @ Sentry

Hey I’m Abhijeet (he/him)

I currently work at Sentry

I help maintain Sentry’s JavaScript SDKs



lforst test(e2e): Add E2E test for sourcemap processing pipeline with deb... … c9aaf8b yesterday 7,652 commits

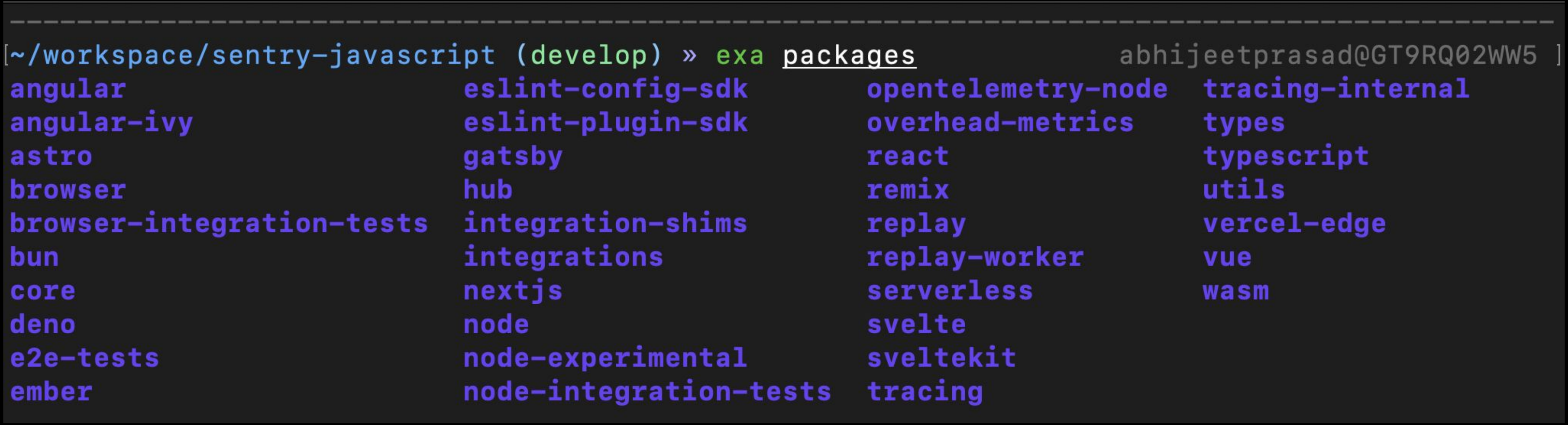
.github	test(e2e): Add E2E test for sourcemap processing pipeline with deb...	yesterday
.vscode	feat: Deno SDK (#9206)	2 days ago
docs	feat(node-experimental): Add `@sentry/node-experimental` packa...	3 months ago
jest	ref(build): Remove `constToVarPlugin` (#5970)	last year
packages	test(e2e): Add E2E test for sourcemap processing pipeline with deb...	yesterday
rollup	feat(astro): Add `sentryAstro` integration (#9218)	yesterday
scenarios/browser	chore(deps-dev): bump webpack in /scenarios/browser	7 months ago
scripts	feat(astro): Add `sentryAstro` integration (#9218)	yesterday
tsconfig-templates	build: Remove `test/tsconfig.json` files (#7606)	7 months ago
vite	test(sveltekit): Switch to vitest (#7438)	7 months ago
.craft.yml	chore(astro): Add Astro package to Craft NPM targets (#9233)	yesterday
.editorconfig	feat: Prepare mono repo, new SDKs (#1349)	5 years ago
.eslintrc.js	chore(various): Fix comments and docstrings (#5651)	last year
.git-blame-ignore-revs	chore: Add test rename to git blame ignore (#7467)	7 months ago

About

Official Sentry SDKs for JavaScript

sentry.io

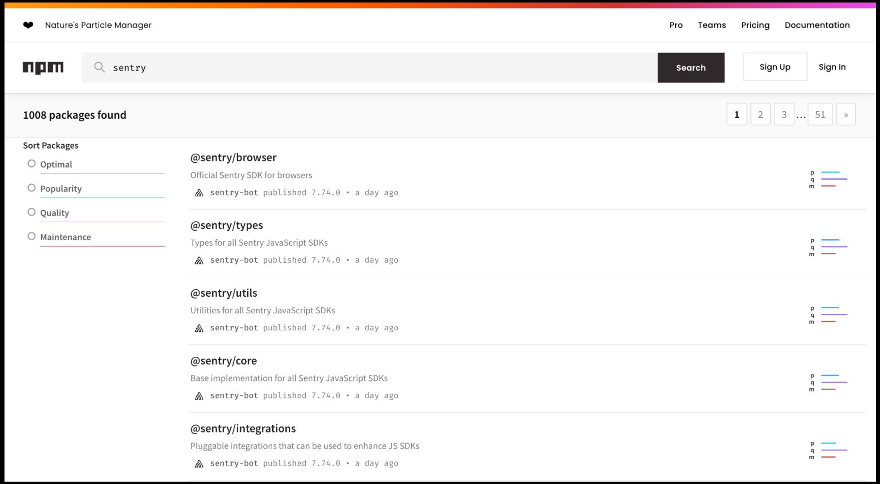
We do a lot at Sentry (20+ JS related SDKs)



```
--------------------------------------------------------------------------------------------
[~/workspace/sentry-javascript (develop) » exa packages          abhijeetprasad@GT9RQ02WW5 ]
angular                      eslint-config-sdk        opentelemetry-node  tracing-internal
angular-ivy                  eslint-plugin-sdk        overhead-metrics    types
astro                        gatsby                   react               typescript
browser                      hub                      remix               utils
browser-integration-tests    integration-shims        replay              vercel-edge
bun                          integrations             replay-worker       vue
core                         nextjs                   serverless          wasm
deno                         node                     svelte
e2e-tests                    node-experimental        sveltekit
ember                        node-integration-tests   tracing
```

How to publish a package!

1. Make sure namespace is free on https://www.npmjs.com/
2. Create a *package.json* file and point it to your JS module via an entrypoint
3. Use *npm cli* to publish your package to npm!
4. Profit!



1008 packages found

@sentry/browser
Official Sentry SDK for browsers
sentry-bot published 7.74.0 • a day ago

@sentry/types
Types for all Sentry JavaScript SDKs
sentry-bot published 7.74.0 • a day ago

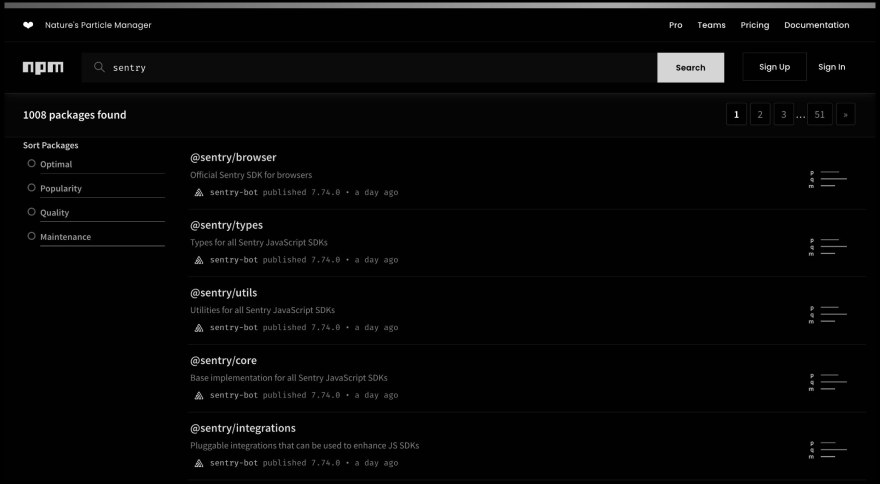
@sentry/utils
Utilities for all Sentry JavaScript SDKs
sentry-bot published 7.74.0 • a day ago

@sentry/core
Base implementation for all Sentry JavaScript SDKs
sentry-bot published 7.74.0 • a day ago

@sentry/integrations
Pluggable integrations that can be used to enhance JS SDKs
sentry-bot published 7.74.0 • a day ago

How to publish a package!

1. Make sure namespace is free on https://www.npmjs.com/
2. Create a *package.json* file and point it to your library via an entrypoint
3. Use *npm cli* to publish your package to npm!
4. Profit!



1008 packages found

@sentry/browser
Official Sentry SDK for browsers
sentry-bot published 7.74.0 • a day ago

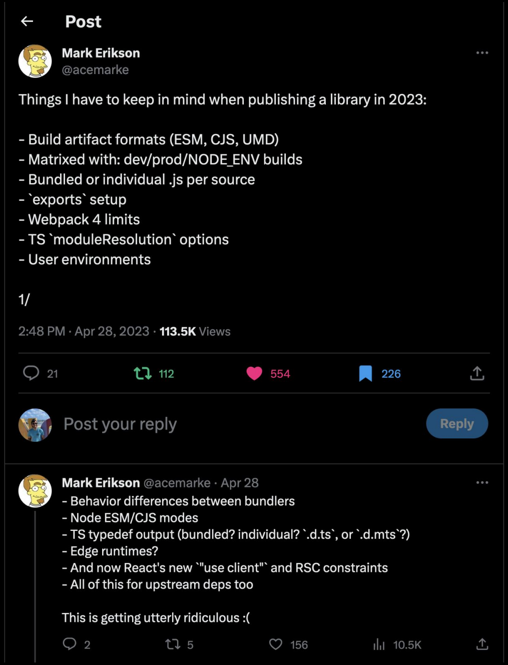
@sentry/types
Types for all Sentry JavaScript SDKs
sentry-bot published 7.74.0 • a day ago

@sentry/utils
Utilities for all Sentry JavaScript SDKs
sentry-bot published 7.74.0 • a day ago

@sentry/core
Base implementation for all Sentry JavaScript SDKs
sentry-bot published 7.74.0 • a day ago

@sentry/integrations
Pluggable integrations that can be used to enhance JS SDKs
sentry-bot published 7.74.0 • a day ago

It’s hard to publish robust and extensible libraries


Post

Mark Erikson
@acemarke

Things I have to keep in mind when publishing a library in 2023:

- Build artifact formats (ESM, CJS, UMD)
- Matrixed with: dev/prod/NODE_ENV builds
- Bundled or individual .js per source
- `exports` setup
- Webpack 4 limits
- TS `moduleResolution` options
- User environments

1/

2:48 PM · Apr 28, 2023 · **113.5K** Views

21 112 554 226

Post your reply Reply

Mark Erikson @acemarke · Apr 28
- Behavior differences between bundlers
- Node ESM/CJS modes
- TS typedef output (bundled? individual? `.d.ts`, or `.d.mts`?)
- Edge runtimes?
- And now React's new `"use client"` and RSC constraints
- All of this for upstream deps too

This is getting utterly ridiculous :(

2 5 156 10.5K

https://twitter.com/acemarke

The JavaScript landscape is pretty big!

Browsers

Runtimes

Frameworks

Things get complicated

- Multiple runtimes
- ESM/CJS/UMD
- JSX, Compilers, Bundlers
- TypeScript
- Tree shaking and bundling
- Deps/Dev Deps/Peer Deps
- Sourcemaps
- Docs and changelogs
- Licensing
- Versioning and LTS
- Directives like “use client”

Today we look at

1. Accounting for different JS runtimes
2. Bundling and module formats
3. TypeScript and publishing types
4. Package health - licensing, versioning, security concerns

A lot of this is high level!

JavaScript Runtimes

Publishing Libraries for the Browser

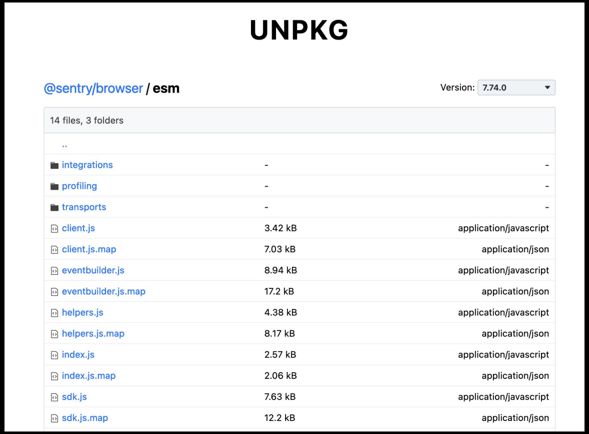
- Have JavaScript version requirements (ES6, ES2020 etc.)
- If you do any transformation (minify/bundle), emit sourcemaps
- Default to emitting ESM if you can
- Make sure to use *files* field or *.npmignore* to only publish what is necessary
- Don’t bundle dependencies not required (use dev and peer deps)

Let’s look at some examples!

Should I bundle?

It depends…

- Better for treeshaking if you keep individual files
- Nice for CDN or unpkg users to have pre-bundled and minified files

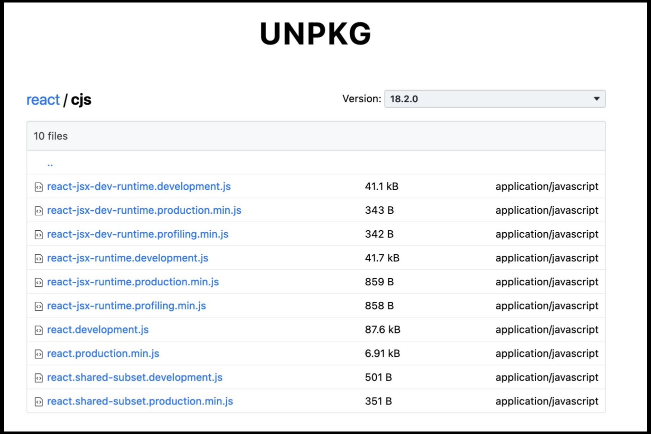


UNPKG

@sentry/browser / esm — Version: 7.74.0

14 files, 3 folders

..		
integrations	-	-
profiling	-	-
transports	-	-
client.js	3.42 kB	application/javascript
client.js.map	7.03 kB	application/json
eventbuilder.js	8.94 kB	application/javascript
eventbuilder.js.map	17.2 kB	application/json
helpers.js	4.38 kB	application/javascript
helpers.js.map	8.17 kB	application/json
index.js	2.57 kB	application/javascript
index.js.map	2.06 kB	application/json
sdk.js	7.63 kB	application/javascript
sdk.js.map	12.2 kB	application/json



UNPKG

react / cjs — Version: 18.2.0

10 files

..		
react-jsx-dev-runtime.development.js	41.1 kB	application/javascript
react-jsx-dev-runtime.production.min.js	343 B	application/javascript
react-jsx-dev-runtime.profiling.min.js	342 B	application/javascript
react-jsx-runtime.development.js	41.7 kB	application/javascript
react-jsx-runtime.production.min.js	859 B	application/javascript
react-jsx-runtime.profiling.min.js	858 B	application/javascript
react.development.js	87.6 kB	application/javascript
react.production.min.js	6.91 kB	application/javascript
react.shared-subset.development.js	501 B	application/javascript
react.shared-subset.production.min.js	351 B	application/javascript

Non-Browser Runtimes

- There are many competing server runtimes for JavaScript

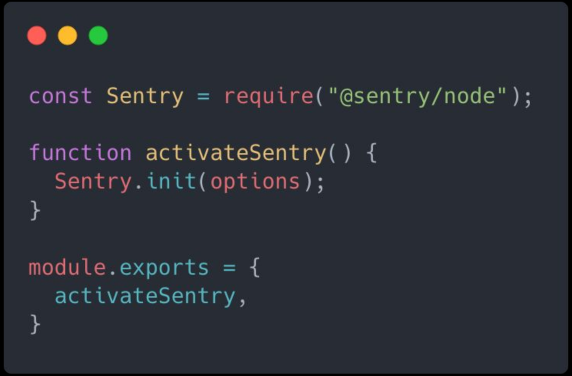
    Node.js, Deno, Cloudflare Workers, Vercel Edge, Bun

- There's also desktop/mobile/embedded runtimes
- Some of these runtimes follow WinterCG common spec, but not all
- If you require something runtime specific - BE CLEAR ABOUT IT

Same rules as publishing for the browser except you might want to think about ESM vs. CJS.

ES Modules (ESM) vs. Common JS Modules (CJS)

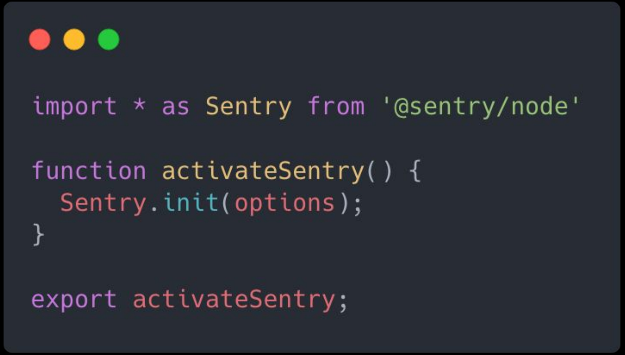
- Two different module mechanisms
- Frontend Frameworks + bundlers -> ESM
- Node had CJS first, now supports ESM
- To enable ESM for node, use *type: module* or use *.mjs* file extension



```
const Sentry = require("@sentry/node");

function activateSentry() {
  Sentry.init(options);
}

module.exports = {
  activateSentry,
}
```



```
import * as Sentry from '@sentry/node'

function activateSentry() {
  Sentry.init(options);
}

export activateSentry;
```

ESM and CJS have incompatibilities

- You can’t use ESM in CJS
    - ESM imports are asynchronous, CJS imports are synchronous
- ESM is Node 12+, CJS doesn’t work in browsers
- ESM does not support monkeypatching
    - There does exist ESM loaders, but this is still experimental API

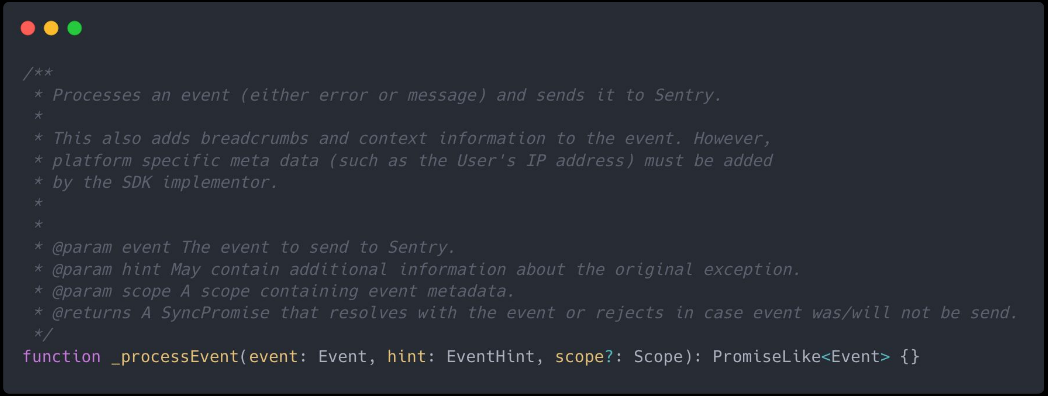
This means you might have to publish both ESM and CJS (watch out for Dual module hazard)

Thing get more complicated when types get involved

TypeScript

TypeScript

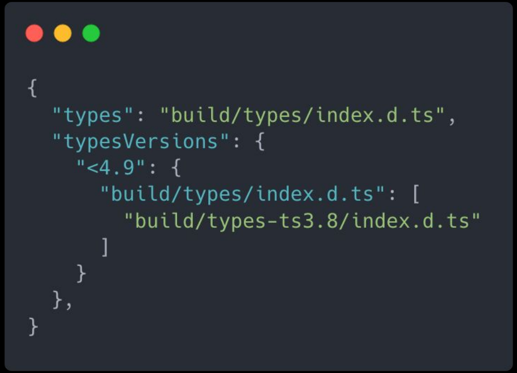
- TypeScript can improve the developer experience of your library
- Two options: write Typescript or use JSDoc
- TS means publishing your types - you can choose where though



```
/**
 * Processes an event (either error or message) and sends it to Sentry.
 *
 * This also adds breadcrumbs and context information to the event. However,
 * platform specific meta data (such as the User's IP address) must be added
 * by the SDK implementor.
 *
 *
 * @param event The event to send to Sentry.
 * @param hint May contain additional information about the original exception.
 * @param scope A scope containing event metadata.
 * @returns A SyncPromise that resolves with the event or rejects in case event was/will not be send.
 */
function _processEvent(event: Event, hint: EventHint, scope?: Scope): PromiseLike<Event> {}
```

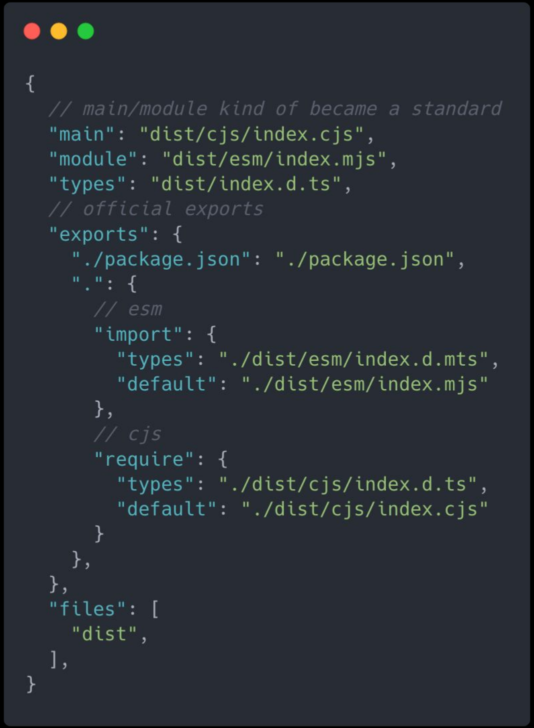
Publishing TypeScript Types

- Decide on DefinitelyTyped or publishing within your own library
- Make sure to publish TS declaration files instead of raw TS
- Use https://arethetypeswrong.github.io/ by @andrewbranch to check if everything is published properly
- You might want to downlevel your types https://github.com/sandersn/downlevel-dts



```json
{
  "types": "build/types/index.d.ts",
  "typesVersions": {
    "<4.9": {
      "build/types/index.d.ts": [
        "build/types-ts3.8/index.d.ts"
      ]
    }
  },
}
```

A standard setup looks something like this.

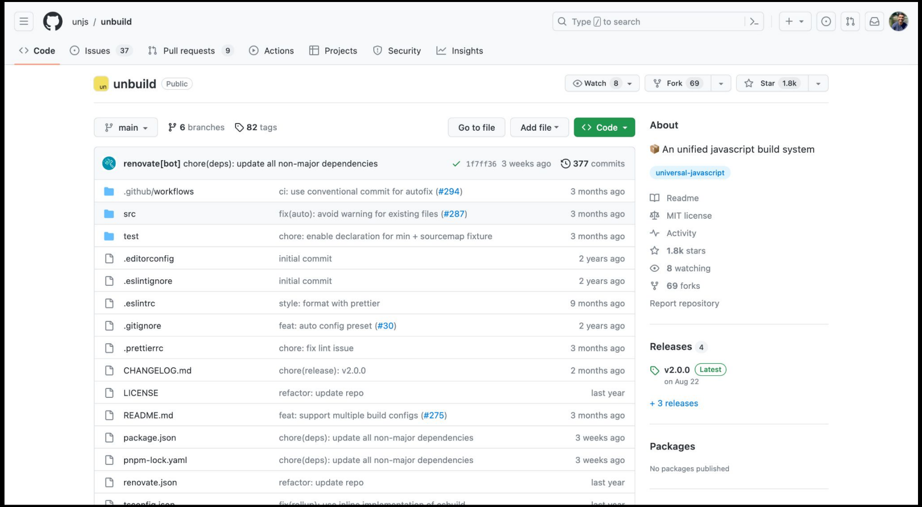


```
{
  // main/module kind of became a standard
  "main": "dist/cjs/index.cjs",
  "module": "dist/esm/index.mjs",
  "types": "dist/index.d.ts",
  // official exports
  "exports": {
    "./package.json": "./package.json",
    ".": {
      // esm
      "import": {
        "types": "./dist/esm/index.d.mts",
        "default": "./dist/esm/index.mjs"
      },
      // cjs
      "require": {
        "types": "./dist/cjs/index.d.ts",
        "default": "./dist/cjs/index.cjs"
      }
    },
  },
  "files": [
    "dist",
  ],
}
```

I recommend using unbuild

https://github.com/unjs/unbuild

- Generates ESM/CJS and puts them in the right places
- Allows you to easily check your subpath exports and conditional exports



renovate[bot] chore(deps): update all non-major dependencies ✓ 1f7ff36 3 weeks ago 377 commits

.github/workflows	ci: use conventional commit for autofix (#294)	3 months ago
src	fix(auto): avoid warning for existing files (#287)	3 months ago
test	chore: enable declaration for min + sourcemap fixture	3 months ago
.editorconfig	initial commit	2 years ago
.eslintignore	initial commit	2 years ago
.eslintrc	style: format with prettier	9 months ago
.gitignore	feat: auto config preset (#30)	2 years ago
.prettierrc	chore: fix lint issue	3 months ago
CHANGELOG.md	chore(release): v2.0.0	2 months ago
LICENSE	refactor: update repo	last year
README.md	feat: support multiple build configs (#275)	3 months ago
package.json	chore(deps): update all non-major dependencies	3 weeks ago
pnpm-lock.yaml	chore(deps): update all non-major dependencies	3 weeks ago
renovate.json	refactor: update repo	last year

About

An unified javascript build system

universal-javascript

Readme
MIT license
Activity
1.8k stars
8 watching
69 forks
Report repository

Releases 4

v2.0.0 Latest
on Aug 22

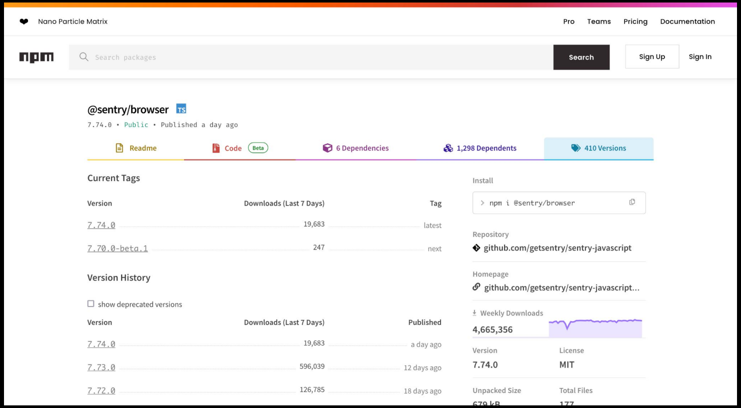
+ 3 releases

Packages

No packages published

Semver and Changelogs

- MAJOR.MINOR.PATCH
- Decide on a versioning scheme to follow
- Have a public changelog
- Libraries like https://github.com/semantic-release/semantic-release can help



@sentry/browser TS

7.74.0 • Public • Published a day ago

Current Tags

Version	Downloads (Last 7 Days)	Tag
7.74.0	19,683	latest
7.70.0-beta.1	247	next

Version History

☐ show deprecated versions

Version	Downloads (Last 7 Days)	Published
7.74.0	19,683	a day ago
7.73.0	596,039	12 days ago
7.72.0	126,785	18 days ago

Install

```
> npm i @sentry/browser
```

Repository
github.com/getsentry/sentry-javascript

Homepage
github.com/getsentry/sentry-javascript...

Weekly Downloads
4,665,356

Version	License
7.74.0	MIT

Unpacked Size
Total Files

Licensing and Docs

- If you want people to use what you've built (and it's open source), always add a LICENSE
- https://choosealicense.com/ can help with this
- Clear READMEs and contributing docs can help with contributors
- If you set up JSDoc or TypeScript can auto-generate docs from code

Easy library publishing?

Making your intentions clear

Intentions?

1. Clearly outlining requirements for JS version and runtimes
2. Defining all your package entry points
3. Stating if package has side effects
4. Emitting sourcemaps
5. Following semver and having a update-to-date changelog
6. Having license and contributing guidelines

You'll run into hurdles - but thats JavaScript for you, enjoy the ride 😄

Thank you!

Twitter: https://twitter.com/imabhiprasad

Bluesky: https://bsky.app/profile/abhiprasad.bsky.social

GitHub: https://github.com/abhiprasad

Open Source JavaScript SDKs: https://github.com/getsentry/sentry-javascript