



# Esbuild

An extremely fast bundler for the web



# What is esbuild?

It is an extremely fast JavaScript and CSS bundler and minifier. Current build tools for the web are 10-100x slower than they could be. The main goal of this project is to bring about a new era of build tool performance, and create an easy-to-use modern bundler along the way

- It developed by Evan Wallace, creator of Figma
- Written in Go
- First release: 2020
- 36.5k stars sur github

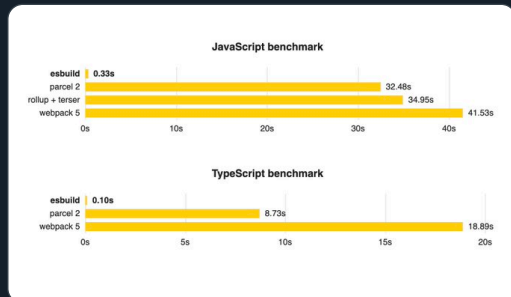
# Benchmark



Evan Wallace | @evanw@hachyderm.io  
@evanwallace · Follow



Huge congrats to the Parcel team for such a big release! I updated esbuild's benchmark results again now that Parcel 2 has come out of beta. The latest release of Parcel 2 is even faster than the beta version and is now faster than Rollup + Terser:



8:16 PM · Oct 14, 2021



441 Reply Copy link

Read 6 replies



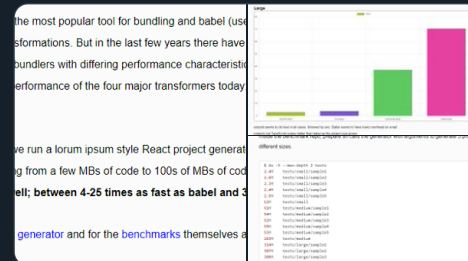
Multiprocess Labs  
@multiprocessio · Follow



Ever been curious about the performance of newer JavaScript transformation tools like esbuild and swc compared to babel and TypeScript?

esbuild and swc lead the pack by a large margin.

[datastition.multiprocess.io/blog/2021-11-1...](https://datastition.multiprocess.io/blog/2021-11-1...)



7:23 PM · Nov 13, 2021



16 Reply Copy link

Read 3 replies

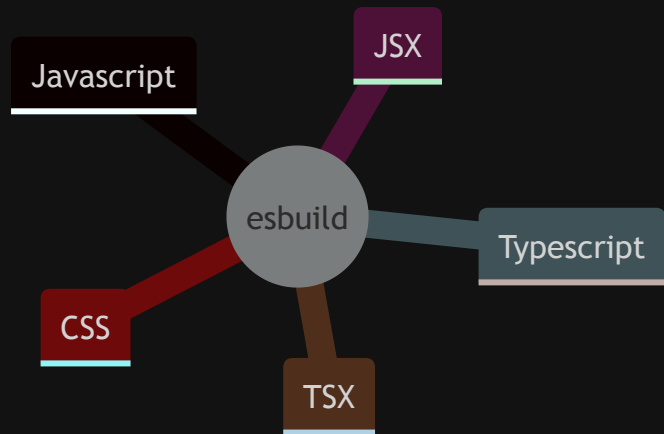
## Performance benchmark for most popular javascript bundlers in various configurations

	Empty	Libraries	Mui	Synthetic
Esbuild	0.046	0.142	0.192	0.685
Parcel: babel + terser	3.737	11.529	8.892	57.232
Rollup: babel + terser	3.121	13.056	9.495	37.689
Rollup: esbuild	1.874	5.553	5.746	14.612
Rollup: swc	1.788	5.966	5.802	14.644
Rspack	0.192	1.308	0.607	5.730
Vite	1.418	6.632	7.957	36.735
Webpack: babel + terser	2.471	11.529	6.406	23.889
Webpack: esbuild	0.808	3.145	2.665	8.798
Webpack: swc	0.849	4.033	2.927	9.134

# Major features

- Extreme speed without needing a cache
- JavaScript, CSS, TypeScript, and JSX built-in
- An API for CLI, JS, and Go
- Plugins
- Bundles ESM and CommonJS modules
- Bundles CSS including CSS modules
- Tree shaking, minification, and source maps
- Local server, watch mode, and plugins

# Who use esbuild?



# Why is it fast?

- It's written in Go and compiles to native code
- Parallelism is used heavily
- Everything in esbuild is written from scratch
- Memory is used efficiently

Each one of these factors is only a somewhat significant speedup, but together they can result in a bundler that is multiple orders of magnitude faster than other bundlers commonly in use today

# A quick start

## CLI mode

```
# Install
npm add -D esbuild

# Add build command to package.json
"scripts": {
  "build-dev": "esbuild src/index.ts --bundle --platform=node --outfile=dist/index.js --format=esm --target=es2022",
  "build": "tsc --noEmit && build-dev",
  "watch": "esbuild src/index.ts --bundle --watch --platform=node --outfile=dist/index.js --format=esm --target=es2022"
}
```

--bundle indicates that it should output only one **file** containing our entire bundle.  
--platform=node indicates that it should bundle **for** Node.js.  
--outfile=dist/index.js indicates that it should output the bundle to dist/index.js.  
--format=esm indicates that it should use **ES** Modules **for** imports **and** exports.

[Learn more about the build options](#)



# Scripts mode

```
# build.js
import esbuild from 'esbuild'
import fg from 'fast-glob'

const entryPoints = fg.sync(['src/**/*.[tj]s'])

esbuild
  .build({
    entryPoints,
    outdir: 'dist',
    platform: 'node',
    sourcemap: true,
    target: 'es2022',
    format: 'esm',
  })
  .catch(() => process.exit(1))
```

# An Extremely Simple React Starter Kit

```
# Project structure
src/
├ components/
│   ├── app.module.css
│   └ app.tsx
├ index.html
├ index.tsx
├ livereload.js
├ style.css
└ types.d.ts

# livereload.js
new EventSource("/esbuild").addEventListener("change", () => location.reload());

# dev server command
esbuild src/index.html src/index.tsx \
  --loader:.html=copy \
  --outdir=build --bundle --watch \
  --servedir=build --serve-fallback=src/index.html \
  --inject:src/livereload.js
```

# Thanks

## Documentations

<https://esbuild.github.io/>

## Github

<https://github.com/evanw/esbuild>

## Benchmark

<https://esbuild.github.io/faq/#benchmark-details>

<https://github.com/s1owjke/js-bundler-benchmark>

## An extremely simple React starter kit

<https://jakelazaroff.com/words/an-extremely-simple-react-starter-kit/>