Rolldown: Chunking in the wild

Jacob Groß – @kurtextrem – 9th October 2025
https://kurtextrem.de

Who am I?

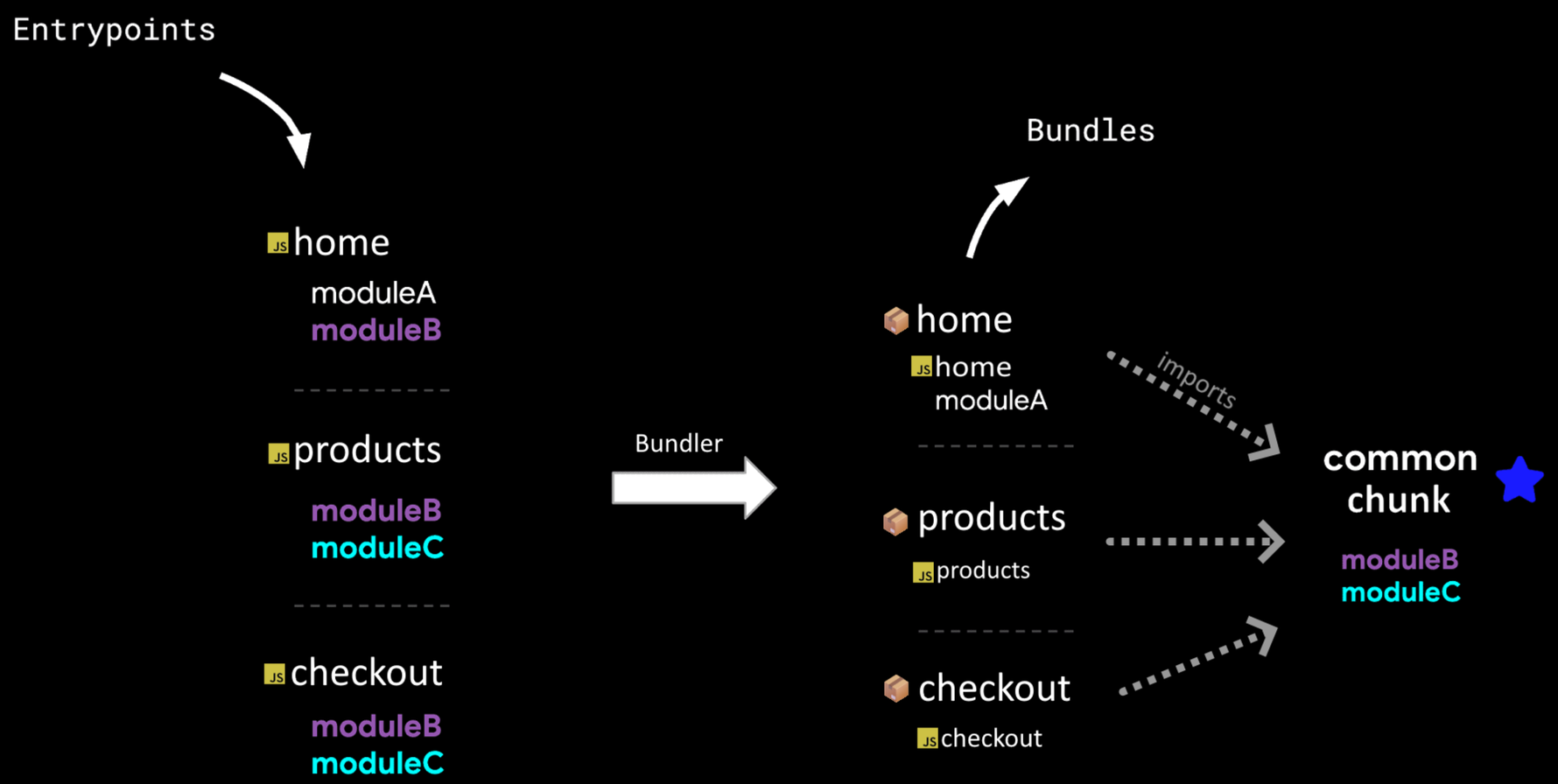
💻 Senior Performance Engineer @ Framer

⚡ I work on Core Web Vitals & performance related topics

🌐 Participant in the W3C WebPerf Working Group – come chat w/ me about perf topics :)

Framer is a website builder and design tool, where you can design any website for any scale, setup complex animations, user interactions, … and have it published in seconds.

What are “chunks”?

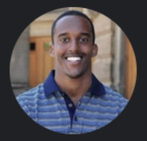


Source: Google / web.dev

How many chunks are optimal?

Improved Next.js and Gatsby page load performance with granular chunking

A newer webpack chunking strategy in Next.js and Gatsby minimizes duplicate code to improve page load performance.

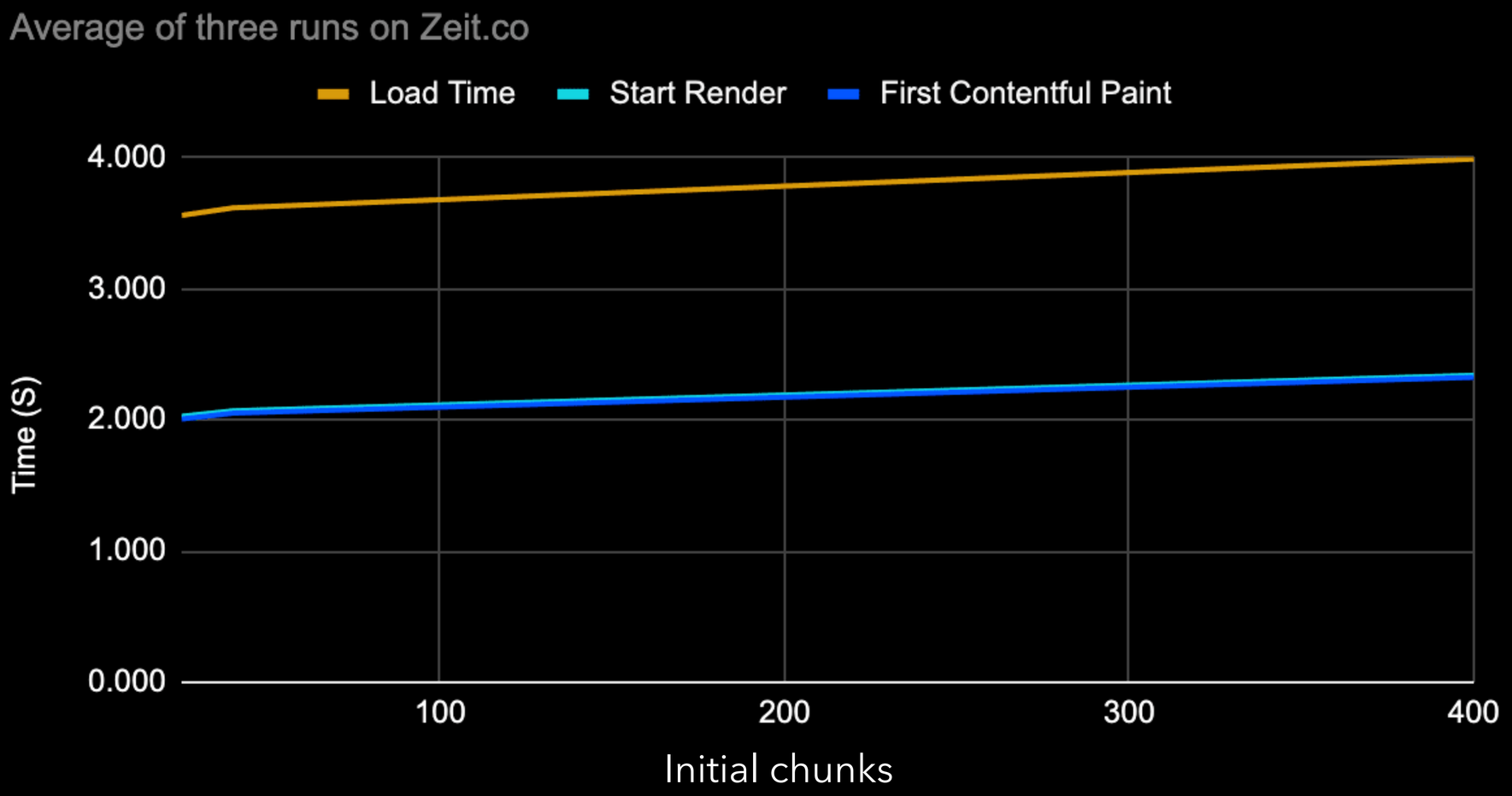


Houssein Djirdeh

Chrome is collaborating with tooling and frameworks in the JavaScript open-source ecosystem. A number of newer optimizations were recently added to improve the loading performance of Next.js and Gatsby. This article covers an improved granular chunking strategy that is now shipped by default in both frameworks.

Source: Google / web.dev

How many chunks are optimal?



Source: Google / web.dev

20-25

“Initial chunks struck the right balance between loading performance and caching efficiency”

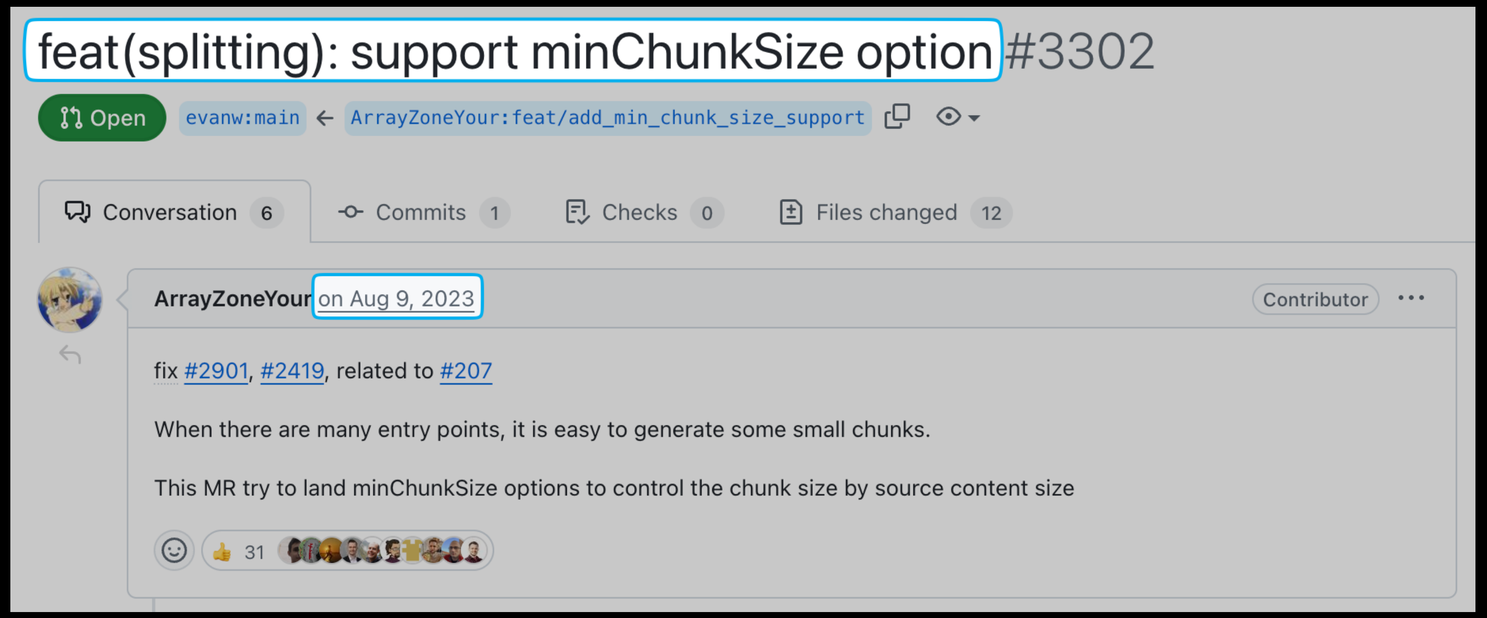
Source: Google / web.dev

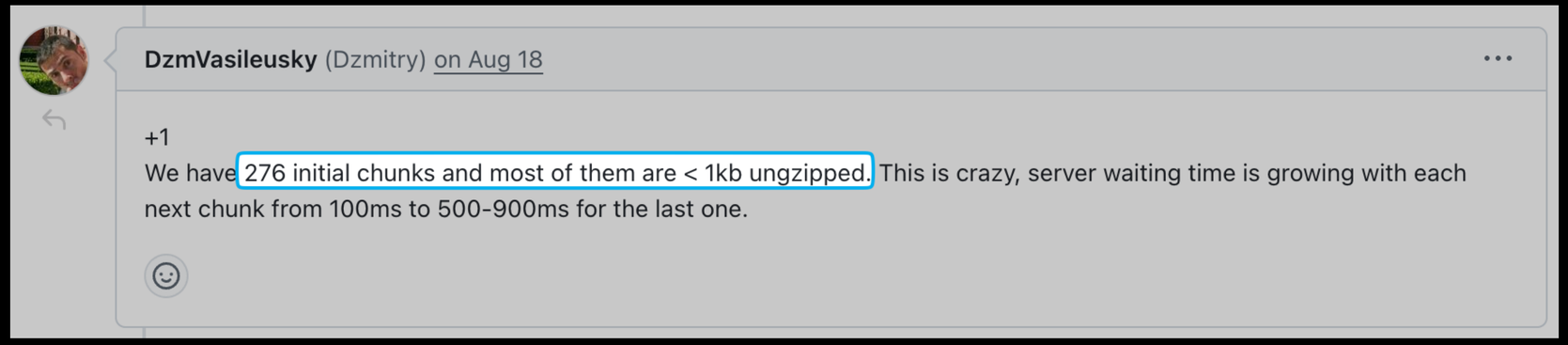
This talk isn’t meant to bash on esbuild.

esbuild served us great. We were (one of?) the first users of esbuild in production. **It’s stable & incredibly fast – Evan Wallace started the race of fast bundlers**.

But: It lacks some options – a void for other bundlers to fill.

Too many small chunks





Not so strict import order

Incorrect import order with code splitting and multiple entry points

Open

iamakulov (Ivan Akulov) opened on Sep 20, 2020 · Edited by iamakulov

Edits

For the workaround, see #399 (comment).

Consider the following code:

```
// entry1.js
import "./init-dep-1.js";
import "./run-dep.js";

// entry2.js
import "./init-dep-2.js";
import "./run-dep.js";

// init-dep-1.js
global.foo = {
  log: () => console.log("foo.log() (from entry 1) called"),
};

// init-dep-2.js
global.foo = {
  log: () => console.log("foo.log() (from entry 2) called"),
};

// run-dep.js
global.foo.log();
```

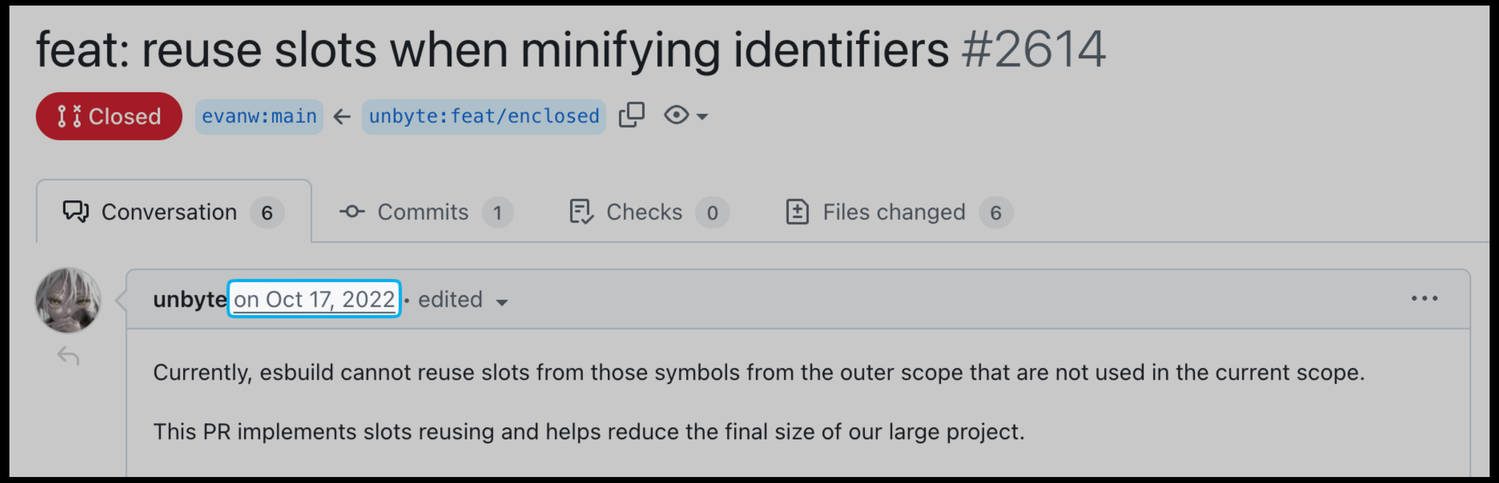
When you bundle this code with `esbuild --bundle --outdir=build --format=esm --splitting --platform=node --out-extension:.js=.mjs ./entry1.js ./entry2.js`, ESBuild discovers that `run-dep.js` is a module shared between both entry points. Because code splitting is enabled, ESBuild moves it into a separate chunk:

```
// build/entry1.js
import "./chunk.P2RPFHF3.js";

global.foo = {
  log: () => console.log("foo.log() (from entry 1) called")
};
```

However, by doing so, ESBuild puts `run-dep.js` above the `init-dep-1.js` or `init-dep-2.js` code. This changes the import

Stale PRs



feat: reuse slots when minifying identifiers #2614

Closed evanw:main ← unbyte:feat/enclosed

Conversation 6 Commits 1 Checks 0 Files changed 6

unbyte on Oct 17, 2022 • edited

Currently, esbuild cannot reuse slots from those symbols from the outer scope that are not used in the current scope.

This PR implements slots reusing and helps reduce the final size of our large project.

Artifact	Original size	Gzip size	
react v17.0.2 (Source)	72.13 kB	19.40 kB	
Minifier	**Minified size**	**Minzipped size**	**Time**
swc	🏆-68% **22.89 kB**	🏆-58% **8.18 kB**	🏆 **22 ms**
terser	-68% 23.14 kB	-57% 8.32 kB	*7x* 153 ms
esbuild-optimized	-67% 23.63 kB	-56% 8.51 kB	*1x* 22 ms
esbuild	-67% 23.70 kB	-56% 8.53 kB	*1x* 22 ms

Alipay *internally* forked esbuild

Fork it?

Unbundle everything?

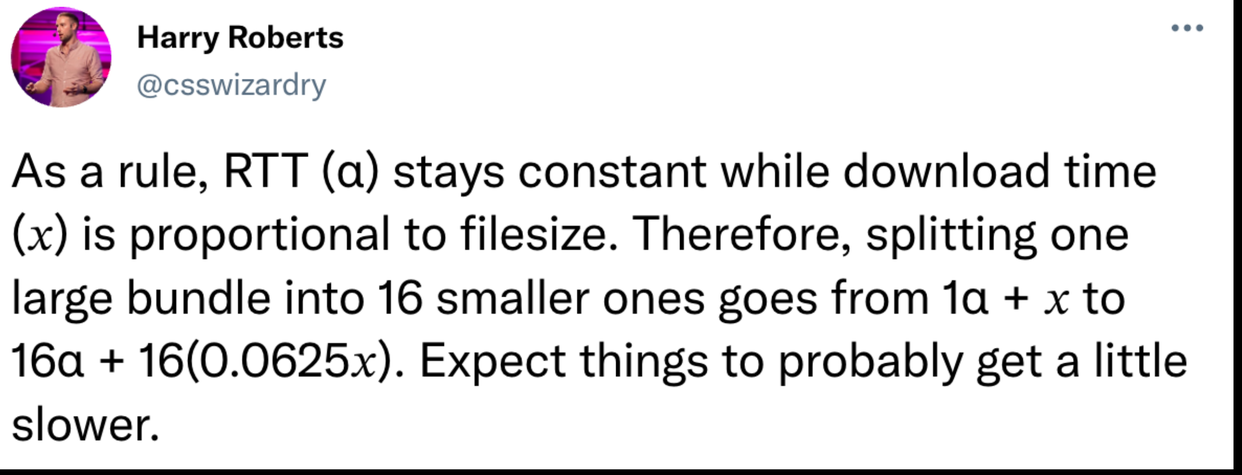
Why bundle in 2025

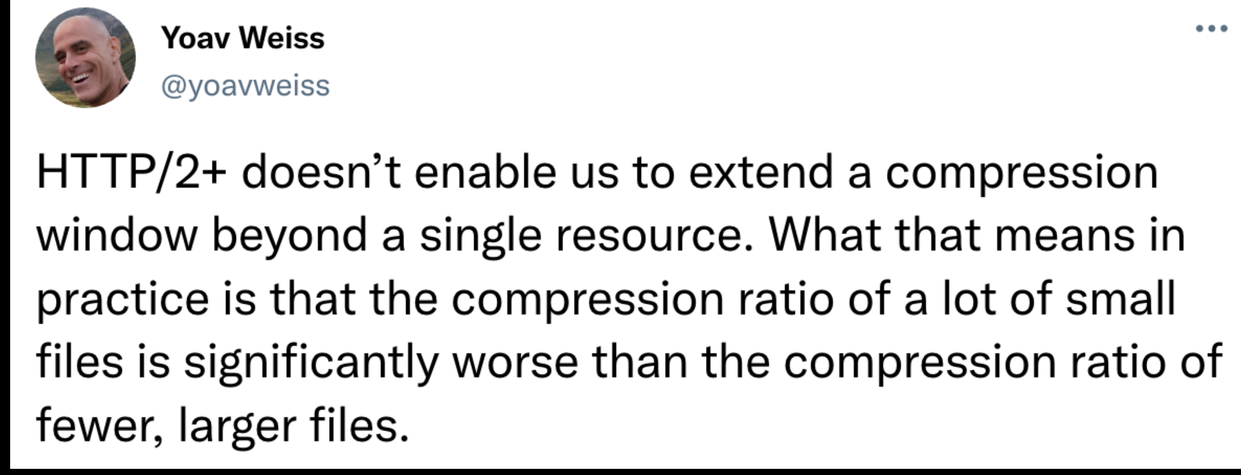
Why bundle?

HTTP/2+ does not solve the need for bundling.

Effects of splitting one file into multiple ones:

Network overhead: ↑ roundtrips, ↓ compression, HTTP header overhead



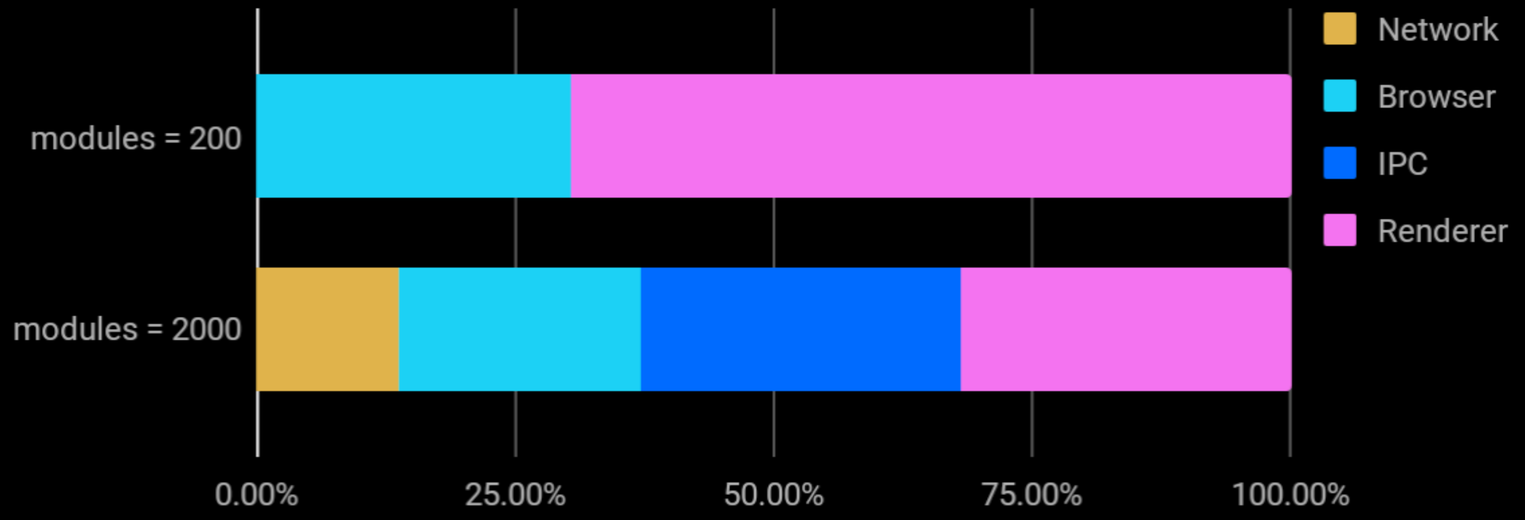


Source: Harry Roberts

Source: Yoav Weiss

Renderer & IPC cost

Modules have a cost in both Blink, V8 and the communication between.

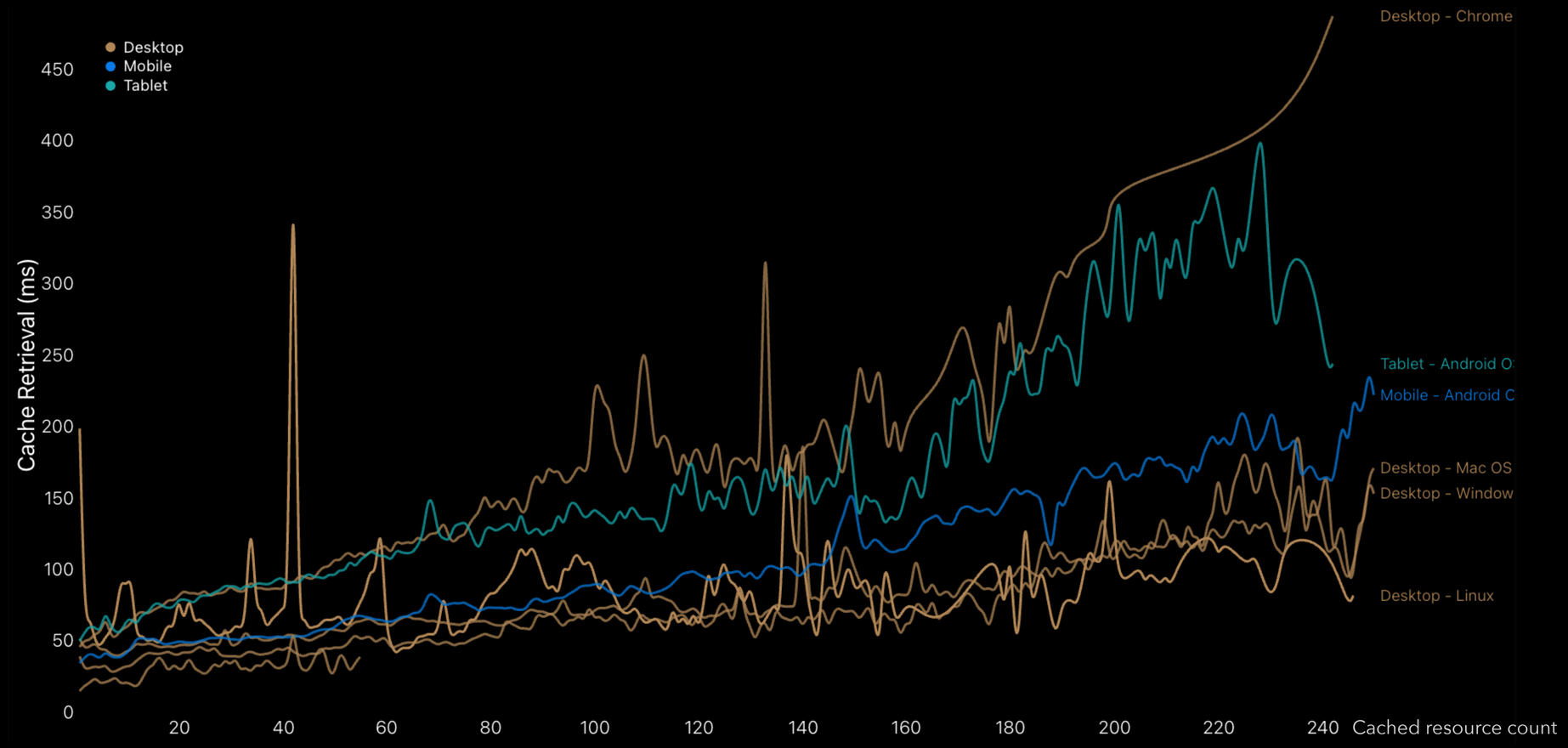


“Loading time increases proportionally to the # of requests, not by the total size of responses, regardless of whether they come from cache or not. For example:

Loading 10_000 1kB files is **10x slower than a single 10MB load**.”

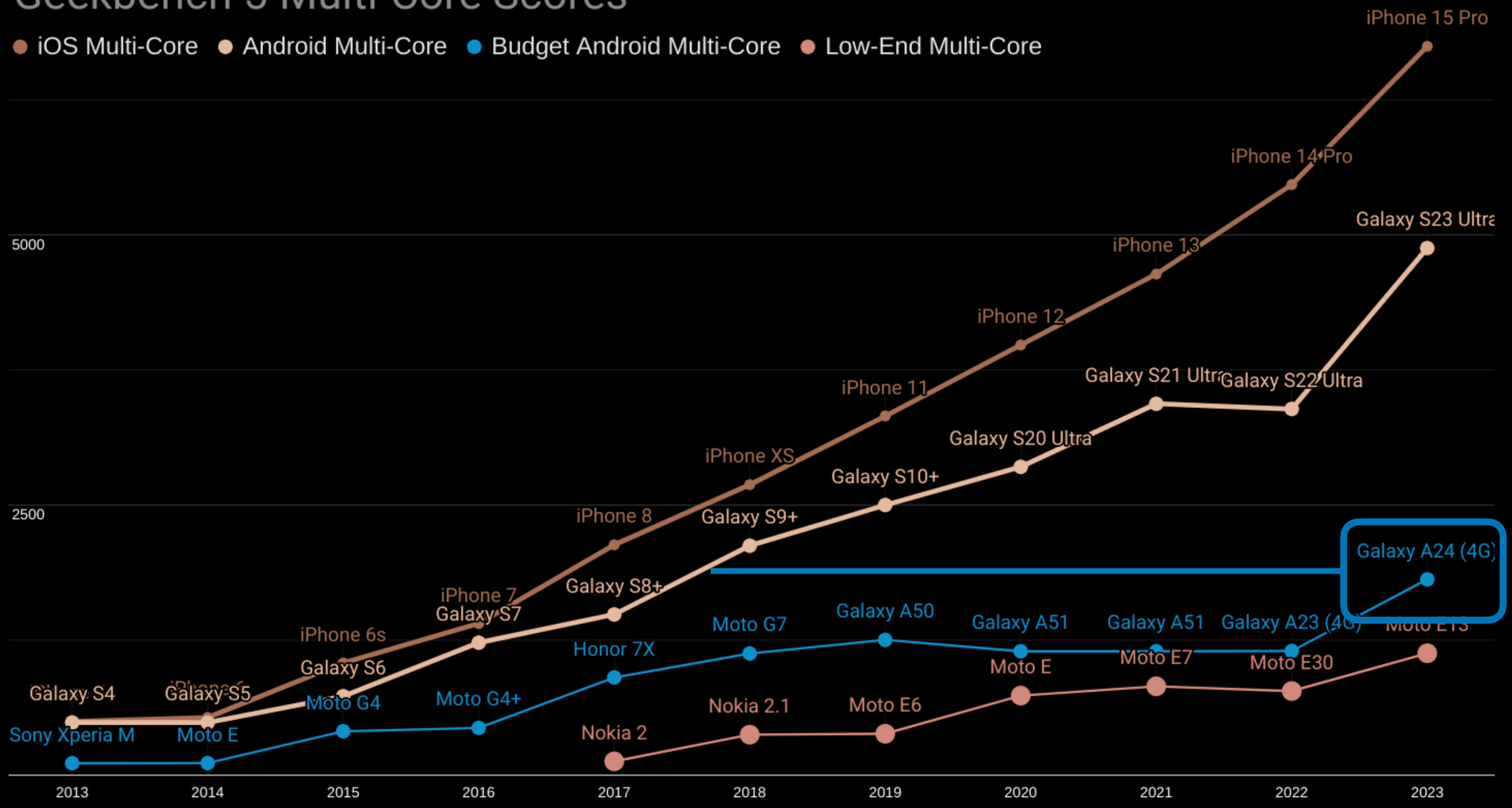
Source: Chromium

Renderer & IPC cost



Chrome OS average cache retrieval **doubles from** ±50ms with **5 cached** resources up **to** ±100ms with **25 resources**.

Source: Simon Hearne



Think about all users

Source: Alex Russell

Concurrency limits

HTTP/2+ servers only accept **N concurrent streams**. After, they kinda behave like HTTP/1.

Table 6.1 Concurrent stream limits on popular HTTP/2 server-side implementations

Software	Type	Default concurrent streams
Apache HTTPD (v2.4.35)	Web server	100[a]
nginx (v1.14.0)	Web server	128[b]
H2O (2.3.0)	Web server	100[c]
IIS (v10)	Web server	100
Jetty (9.4.12)	Web and Java servlet container	128[d]
Apache Tomcat (9.0)	Web and Java servlet container	200[e]
Node (10.11.0)	JavaScript runtime environment	100[f]
Akamai	CDN	100
Amazon CloudFront and S3	CDN	128
Cloudflare	CDN	128
MaxCDN	CDN	128

Table 6.2 Concurrent stream limits on popular HTTP/2 web browsers

Software	Default concurrent streams
Chrome (v69)	1000
Firefox (v62)	Not set (uses HTTP/2 default of unlimited)
Safari (v12)	1000
Opera (v56)	1000
Edge (v17)	1024
Internet Explorer 11	1024

Source: Barry Pollard / "HTTP/2 in action"

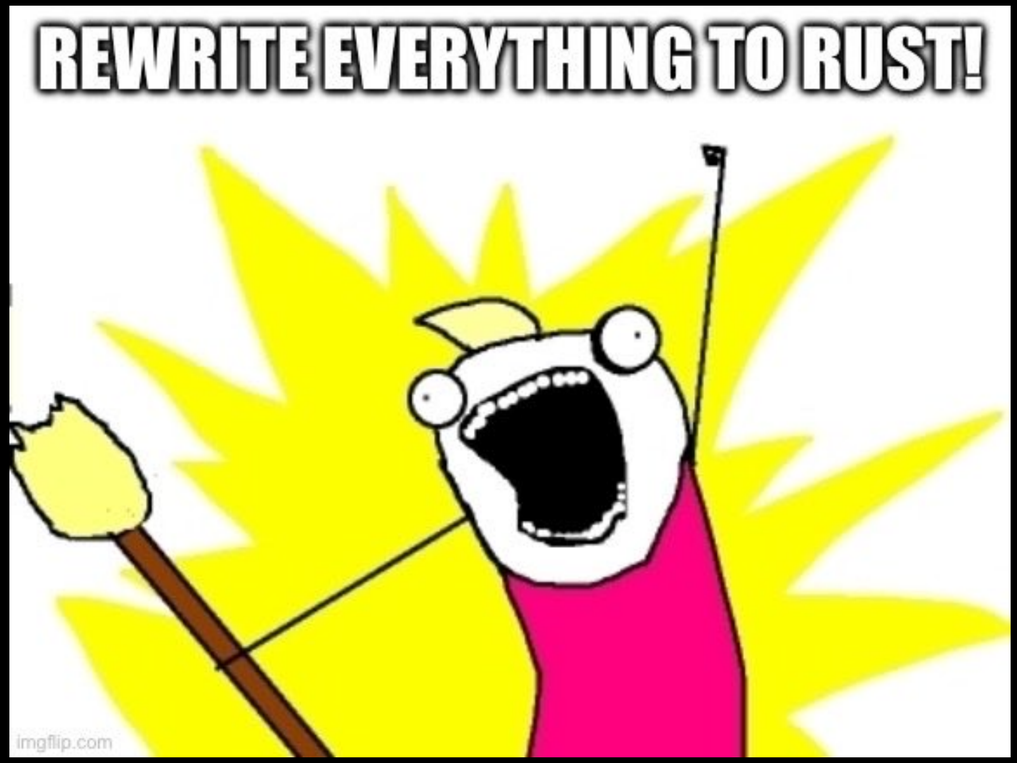
You should bundle.

HTTP/2+ does not solve the need for bundling.

Splitting one file into multiple ones creates more overhead:

1. **Network overhead**: ↑ roundtrips, ↓ compression, HTTP header overhead
2. **Renderer & IPC cost**: More isn't free.
3. **Concurrency limits** of HTTP/2+ servers & browsers

This leads to **worse UX & Core Web Vitals** – especially on slower devices & networks.


REWRITE EVERYTHING TO RUST!
imgflip.com

Ambitious bundler goals

Our 3 must-have requirements:

1. **Granular control over chunks** – to counter too-small-modules.
2. **Strict module execution order** – to isolate user code in our case.
3. **Equally or faster than esbuild** – regressing perf isn’t an option.

A bundler journey

There are other good bundlers out there:

- **Parcel**: Surprisingly fast and capable. Slower than esbuild.
- **Rspack**: Similar API to Webpack. In my tests in 2024, slower than Parcel.
- **Farm-Fe**: Unique chunking algo. Slower than Rspack.
- **Turbopack**: Tied to Next.js.


MISSION IMPOSSIBLE?

😰

Back in May 2024, we knew Rolldown was coming, but it was early.

We’ve seen how fast Oxc is. We heard Rolldown is *fast*.

Rolldown at Framer

from research to product

After talking in August 2024, we figured a cooperation would shape Framer's future of bundling and I could help shape Rolldown's API.

What happened in 1 year?

- Designing & implementing the chunking API
- Minification support
- Fix rare minification errors
- Handle legal comments during minification
- Fix crashes
- Hashing woes
- Top-level await support
- CJS in ESM
- Nasty chunking algorithm things (Yunfei can tell you a story)
- Performance & memory improvements (bundle 300 MB JS?)
- Test Rolldown for Framer sites with 1 route and with 10_000s of routes

Esbuild vs Rolldown

(from my POV)

✅ Similar API (with unique Rolldown additions)

✅ JS plugin support (we use it to create virtual modules)

✅ JS minification

❌ No CSS minification yet!

Minifier performance

Artifact	Original size	Gzip size	
typescript v4.9.5 (Source)	10.95 MB	1.88 MB	
Minifier	**Minified size**	**Minzipped size**	**Time**
1. oxc-minify	-70% 3.34 MB	🏆-55% **855.08 KB**	*5x* 607 ms
2. @tdewolff/minify	-69% 3.35 MB	-54% 875.82 KB	*2x* 255 ms
3. @swc/core	🏆-70% **3.31 MB**	-54% 859.05 KB	*13x* 1,487 ms
4. esbuild	-68% 3.49 MB	-51% 915.59 KB	*4x* 485 ms
5. bun	-68% 3.54 MB	-51% 923.27 KB	*2x* 293 ms
6. uglify-js (no compress)	-68% 3.54 MB	-53% 876.54 KB	*34x* 3,784 ms
7. terser (no compress)	-68% 3.53 MB	-53% 878.64 KB	*43x* 4,687 ms

Esbuild vs Rolldown

Memory usage when bundling 300 MB JS *w/* 500 MB sourcemaps:

- Rolldown: **>10 GB**
- esbuild: **9.2 GB**

Improvements are coming! (And I hope no one else has 300 MB of JS somewhere)

Bundling at Framer

- Customers can write custom JS code, most commonly React components → our biggest sites have ±*300 MB of JS* (across all pages)

- We use SSG for faster page loads & crawl-ability (no SSR → no LLM!) 🤖

- Rolldown bundles both client and server code
  → 🪶 **Deps needed for SSR shouldn’t end up in the client bundle**
  → 🔒 **Strict execution order matters** – our runtime code shouldn’t fail if there is an error in user code

Bundling at Framer

🚪 Every route has its own entry-point
→ **Lazy loaded** in the browser (& preloaded using heuristics)

📦 Every route might have its own dependencies
→ **They shouldn’t be loaded on every other route**

Non trivial: **How do you split the code optimally?**

Good rules for bundling

Google research to the rescue:

A. **Used by some chunks and below 20 kb**
   → “common” chunks

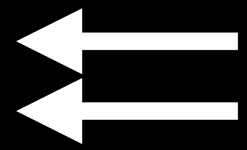
B. **Used on all pages**
   → “shared” chunks

C. **Big library > 160kb**
   → “library” chunks

Source: Google / web.dev

Start simple

```
groups: [
    {
        name: "react",
        // include react/react-dom/scheduler, but not the server chunk
        test: /[\\/]npm:(react|react-dom|scheduler)@(?![\d.]+\/server\.browser\.js)/,
        priority: 1000,
    },
    { name: "motion", test: /^(motion|framer-motion)/, },
    { name: "framer", test: /^framer/, },
    {
        name: "shared-lib",
        minShareCount: routesLength,
    },
    routesLength >= 40 ? {
        name: "shared",
        minShareCount: Math.floor(routesLength / 2),
    } : undefined,
].filter(group => group !== undefined),
```



Update independently

Used by all routes

More likely to hit

20-25 chunks

Advanced Options in Rolldown

Goal: Simple, yet powerful options similar to webpack.

A. **`minSize`** → don't create the group unless > N bytes

B. **`maxSize`** → split the group when > N bytes

C. **`minShareCount`**
   → create only if N entrypoints reference it

D. **`min/maxModuleSize`**
   → min or max size of modules that are bundled into the group

How Framer loads JS

```
<body>
    <div id="main"><!-- react root --></div>
    ...
    <link rel="modulepreload" href="chunk1.mjs">
    <link rel="modulepreload" href="chunk2.mjs">
    <script type="module" src="main.mjs" async></script>
</body>
```

Using **HTTP/3**
from our CDN

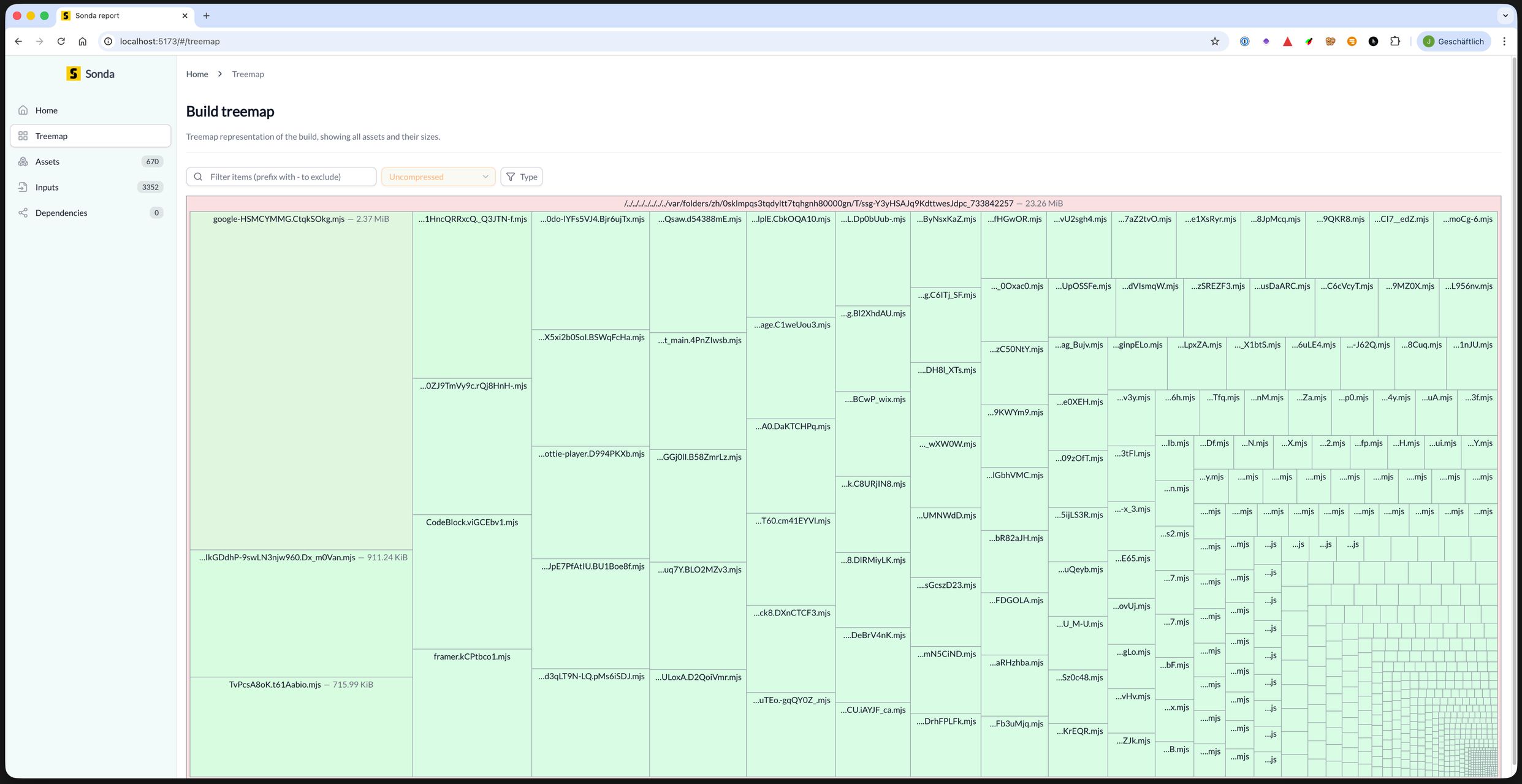
Browser preload scanner starts downloading the chunks before executing “main”

Analyze your chunks

Analyzing bundles

- *Spoiler: Bundle analysis will be part of Vite Devtools, stay tuned for Anthony's talk*

- For pure Rolldown users, currently there are two plugins:
  - Sonda
  - Rollup-plugin-visualizer
- DevTools is a viable option too


Sonda report
localhost:5173/#/treemap
Geschäftlich
Sonda
Home
Treemap
Assets 670
Inputs 3352
Dependencies 0
Home > Treemap
Build treemap
Treemap representation of the build, showing all assets and their sizes.
Filter items (prefix with - to exclude)
Uncompressed
Type
/../../../../../../var/folders/zh/0skImpqs3tqdyltt7tqhgnh80000gn/T/ssg-Y3yHSAJq9KdttwesJdpc_733842257 — 23.26 MiB
google-HSMCYMMG.CtqkSOkg.mjs — 2.37 MiB
...IkGDdhP-9swLN3njw960.Dx_m0Van.mjs — 911.24 KiB
TvPcsA8oK.t61Aabio.mjs — 715.99 KiB
...1HncQRRxcQ._Q3JTN-f.mjs
...0ZJ9TmVy9c.rQj8HnH-.mjs
CodeBlock.viGCEbv1.mjs
framer.kCPtbco1.mjs
...0do-IYFs5VJ4.Bjr6ujTx.mjs
...X5xi2b0SoI.BSWqFcHa.mjs
...ottie-player.D994PKXb.mjs
...JpE7PfAtIU.BU1Boe8f.mjs
...d3qLT9N-LQ.pMs6iSDJ.mjs
...Qsaw.d54388mE.mjs
...t_main.4PnZIwsb.mjs
...GGj0II.B58ZmrLz.mjs
...uq7Y.BLO2MZv3.mjs
...ULoxA.D2QoiVmr.mjs
...lplE.CbkOQA10.mjs
...age.C1weUou3.mjs
...A0.DaKTCHPq.mjs
...T60.cm41EYVl.mjs
...ck8.DXnCTCF3.mjs
...uTEo.-gqQY0Z_.mjs
...L.Dp0bUub-.mjs
...g.Bl2XhdAU.mjs
....BCwP_wix.mjs
...k.C8URjIN8.mjs
...8.DIRMiyLK.mjs
....DeBrV4nK.mjs
...CU.iAYJF_ca.mjs
...ByNsxKaZ.mjs
...g.C6ITj_SF.mjs
....DH8l_XTs.mjs
..._wXW0W.mjs
...UMNWdD.mjs
....sGcszD23.mjs
...mN5CiND.mjs
....DrhFPLFk.mjs
...fHGwOR.mjs
..._0Oxac0.mjs
...zC50NtY.mjs
...9KWYm9.mjs
...lGbhVMC.mjs
...bR82aJH.mjs
...FDGOLA.mjs
...aRHzhba.mjs
...Fb3uMjq.mjs
...vU2sgh4.mjs
...UpOSSFe.mjs
...ag_Bujv.mjs
...e0XEH.mjs
...09zOfT.mjs
...5ijLS3R.mjs
...uQeyb.mjs
...U_M-U.mjs
...Sz0c48.mjs
...KrEQR.mjs
...7aZ2tvO.mjs
...dVIsmqW.mjs
...ginpELo.mjs
...v3y.mjs
...3tFl.mjs
...-x_3.mjs
...E65.mjs
...ovUj.mjs
...gLo.mjs
...vHv.mjs
...ZJk.mjs
...e1XsRyr.mjs
...zSREZF3.mjs
...LpxZA.mjs
...6h.mjs
...lb.mjs
...n.mjs
...s2.mjs
...7.mjs
...7.mjs
...bF.mjs
...x.mjs
...B.mjs
...8JpMcq.mjs
...usDaARC.mjs
..._X1btS.mjs
...Tfq.mjs
...Df.mjs
...y.mjs
....mjs
...js
...9QKR8.mjs
...C6cVcyT.mjs
...6uLE4.mjs
...Za.mjs
...N.mjs
...X.mjs
...2.mjs
...fp.mjs
...H.mjs
...ui.mjs
...CI7__edZ.mjs
...9MZ0X.mjs
...-J62Q.mjs
...p0.mjs
...4y.mjs
...uA.mjs
...Y.mjs
...3f.mjs
...moCg-6.mjs
...L956nv.mjs
...8Cuq.mjs
...1nJU.mjs

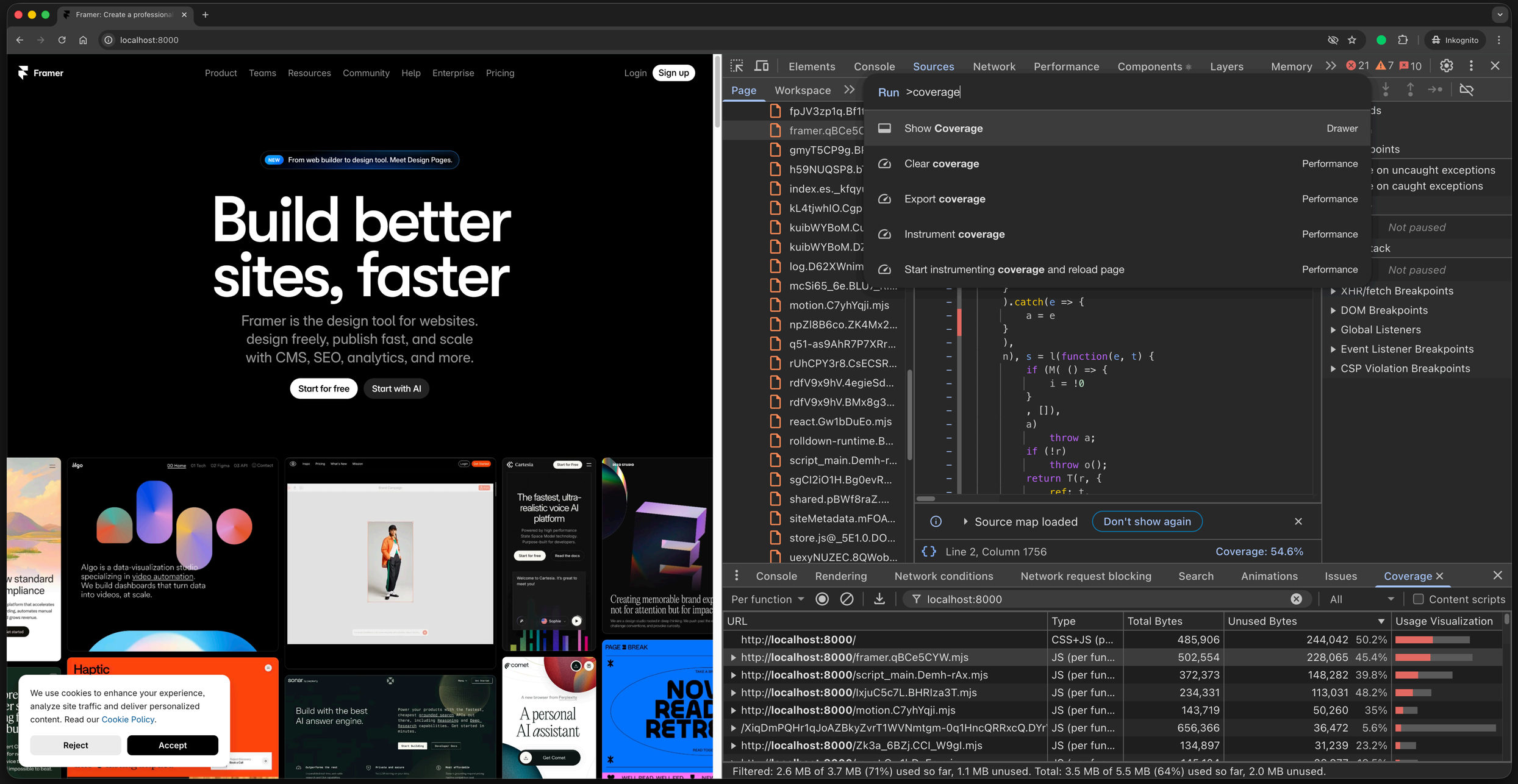
Sonda

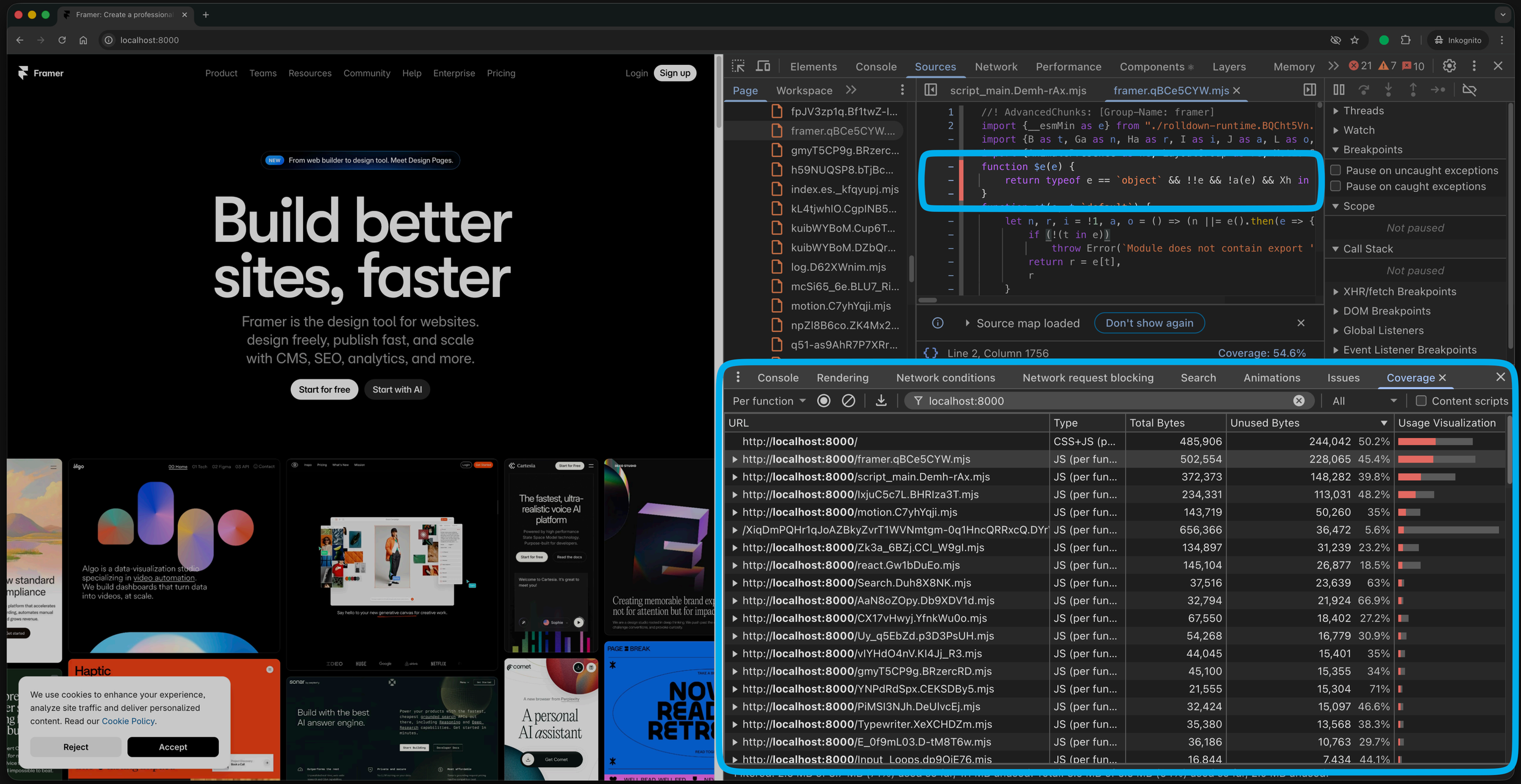
Home
Treemap
Assets 670
Inputs 3352
Dependencies 0

Build treemap

Treemap representation of the build, showing all assets and their sizes.

codemirror | Uncompressed | Type

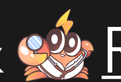

Framer: Create a professional
localhost:8000
Inkognito
Framer
Product Teams Resources Community Help Enterprise Pricing
Login Sign up
NEW From web builder to design tool. Meet Design Pages.
Build better sites, faster
Framer is the design tool for websites. design freely, publish fast, and scale with CMS, SEO, analytics, and more.
Start for free
Start with AI
Algo is a data-visualization studio specializing in video automation. We build dashboards that turn data into videos, at scale.
The fastest, ultra-realistic voice AI platform
Creating memorable brand exp not for attention but for impac
Haptic
We use cookies to enhance your experience, analyze site traffic and deliver personalized content. Read our Cookie Policy.
Reject
Accept
Build with the best AI answer engine.
A personal AI assistant
Get Comet
Elements Console Sources Network Performance Components Layers Memory
Page Workspace
Run >coverage
Show Coverage Drawer
Clear coverage Performance
Export coverage Performance
Instrument coverage Performance
Start instrumenting coverage and reload page Performance
Source map loaded
Don't show again
Line 2, Column 1756
Coverage: 54.6%
XHR/fetch Breakpoints
DOM Breakpoints
Global Listeners
Event Listener Breakpoints
CSP Violation Breakpoints
Console Rendering Network conditions Network request blocking Search Animations Issues Coverage
Per function
localhost:8000
All
Content scripts
URL Type Total Bytes Unused Bytes Usage Visualization
http://localhost:8000/ CSS+JS (p... 485,906 244,042 50.2%
http://localhost:8000/framer.qBCe5CYW.mjs JS (per fun... 502,554 228,065 45.4%
http://localhost:8000/script_main.Demh-rAx.mjs JS (per fun... 372,373 148,282 39.8%
http://localhost:8000/IxjuC5c7L.BHRIza3T.mjs JS (per fun... 234,331 113,031 48.2%
http://localhost:8000/motion.C7yhYqji.mjs JS (per fun... 143,719 50,260 35%
http://localhost:8000/XiqDmPQHr1qJoAZBkyZvrT1WVNmtgm-0q1HncQRRxcQ.DYr JS (per fun... 656,366 36,472 5.6%
http://localhost:8000/Zk3a_6BZj.CCl_W9gI.mjs JS (per fun... 134,897 31,239 23.2%
Filtered: 2.6 MB of 3.7 MB (71%) used so far, 1.1 MB unused. Total: 3.5 MB of 5.5 MB (64%) used so far, 2.0 MB unused.


Framer: Create a professional
localhost:8000
Inkognito
Framer
Product
Teams
Resources
Community
Help
Enterprise
Pricing
Login
Sign up
NEW
From web builder to design tool. Meet Design Pages.
Build better sites, faster
Framer is the design tool for websites. design freely, publish fast, and scale with CMS, SEO, analytics, and more.
Start for free
Start with AI
Algo is a data-visualization studio specializing in video automation. We build dashboards that turn data into videos, at scale.
Say hello to your new generative canvas for creative work.
The fastest, ultra-realistic voice AI platform
Creating memorable brand experiences not for attention but for impact
Haptic
We use cookies to enhance your experience, analyze site traffic and deliver personalized content. Read our Cookie Policy.
Reject
Accept
Build with the best AI answer engine.
A personal AI assistant
Get Comet
Elements
Console
Sources
Network
Performance
Components
Layers
Memory
Page
Workspace
script_main.Demh-rAx.mjs
framer.qBCe5CYW.mjs
fpJV3zp1q.Bf1twZ-I...
framer.qBCe5CYW....
gmyT5CP9g.BRzerc...
h59NUQSP8.bTjBc...
index.es._kfqyupj.mjs
kL4tjwhIO.CgpINB5...
kuibWYBoM.Cup6T...
kuibWYBoM.DZbQr...
log.D62XWnim.mjs
mcSi65_6e.BLU7_Ri...
motion.C7yhYqji.mjs
npZl8B6co.ZK4Mx2...
q51-as9AhR7P7XRr...
//! AdvancedChunks: [Group-Name: framer]
import {__esmMin as e} from "./rolldown-runtime.BQCht5Vn.
import {B as t, Ga as n, Ha as r, I as i, J as a, L as o,
function $e(e) {
return typeof e == `object` && !!e && !a(e) && Xh in
}
let n, r, i = !1, a, o = () => (n ||= e().then(e => {
if (!(t in e))
throw Error(`Module does not contain export '
return r = e[t],
r
}
Source map loaded
Don't show again
Line 2, Column 1756
Coverage: 54.6%
Threads
Watch
Breakpoints
Pause on uncaught exceptions
Pause on caught exceptions
Scope
Not paused
Call Stack
Not paused
XHR/fetch Breakpoints
DOM Breakpoints
Global Listeners
Event Listener Breakpoints
Console
Rendering
Network conditions
Network request blocking
Search
Animations
Issues
Coverage
Per function
localhost:8000
All
Content scripts
URL
Type
Total Bytes
Unused Bytes
Usage Visualization
http://localhost:8000/ CSS+JS (p... 485,906 244,042 50.2%
http://localhost:8000/framer.qBCe5CYW.mjs JS (per fun... 502,554 228,065 45.4%
http://localhost:8000/script_main.Demh-rAx.mjs JS (per fun... 372,373 148,282 39.8%
http://localhost:8000/lxjuC5c7L.BHRIza3T.mjs JS (per fun... 234,331 113,031 48.2%
http://localhost:8000/motion.C7yhYqji.mjs JS (per fun... 143,719 50,260 35%
/XiqDmPQHr1qJoAZBkyZvrT1WVNmtgm-0q1HncQRRxcQ.DYr JS (per fun... 656,366 36,472 5.6%
http://localhost:8000/Zk3a_6BZj.CCl_W9gI.mjs JS (per fun... 134,897 31,239 23.2%
http://localhost:8000/react.Gw1bDuEo.mjs JS (per fun... 145,104 26,877 18.5%
http://localhost:8000/Search.Duh8X8NK.mjs JS (per fun... 37,516 23,639 63%
http://localhost:8000/AaN8oZOpy.Db9XDV1d.mjs JS (per fun... 32,794 21,924 66.9%
http://localhost:8000/CX17vHwyj.YfnkWu0o.mjs JS (per fun... 67,550 18,402 27.2%
http://localhost:8000/Uy_q5EbZd.p3D3PsUH.mjs JS (per fun... 54,268 16,779 30.9%
http://localhost:8000/vIYHdO4nV.KI4Jj_R3.mjs JS (per fun... 44,045 15,401 35%
http://localhost:8000/gmyT5CP9g.BRzercRD.mjs JS (per fun... 45,100 15,355 34%
http://localhost:8000/YNPdRdSpx.CEKSDBy5.mjs JS (per fun... 21,555 15,304 71%
http://localhost:8000/PiMSI3NJh.DeUlvcEj.mjs JS (per fun... 32,424 15,097 46.6%
http://localhost:8000/Typewriter.XeXCHDZm.mjs JS (per fun... 35,380 13,568 38.3%
http://localhost:8000/E_0f9mL03.D-tM8T6w.mjs JS (per fun... 36,186 10,763 29.7%
http://localhost:8000/Input_Loops.dp9OiE76.mjs JS (per fun... 16,844 7,434 44.1%

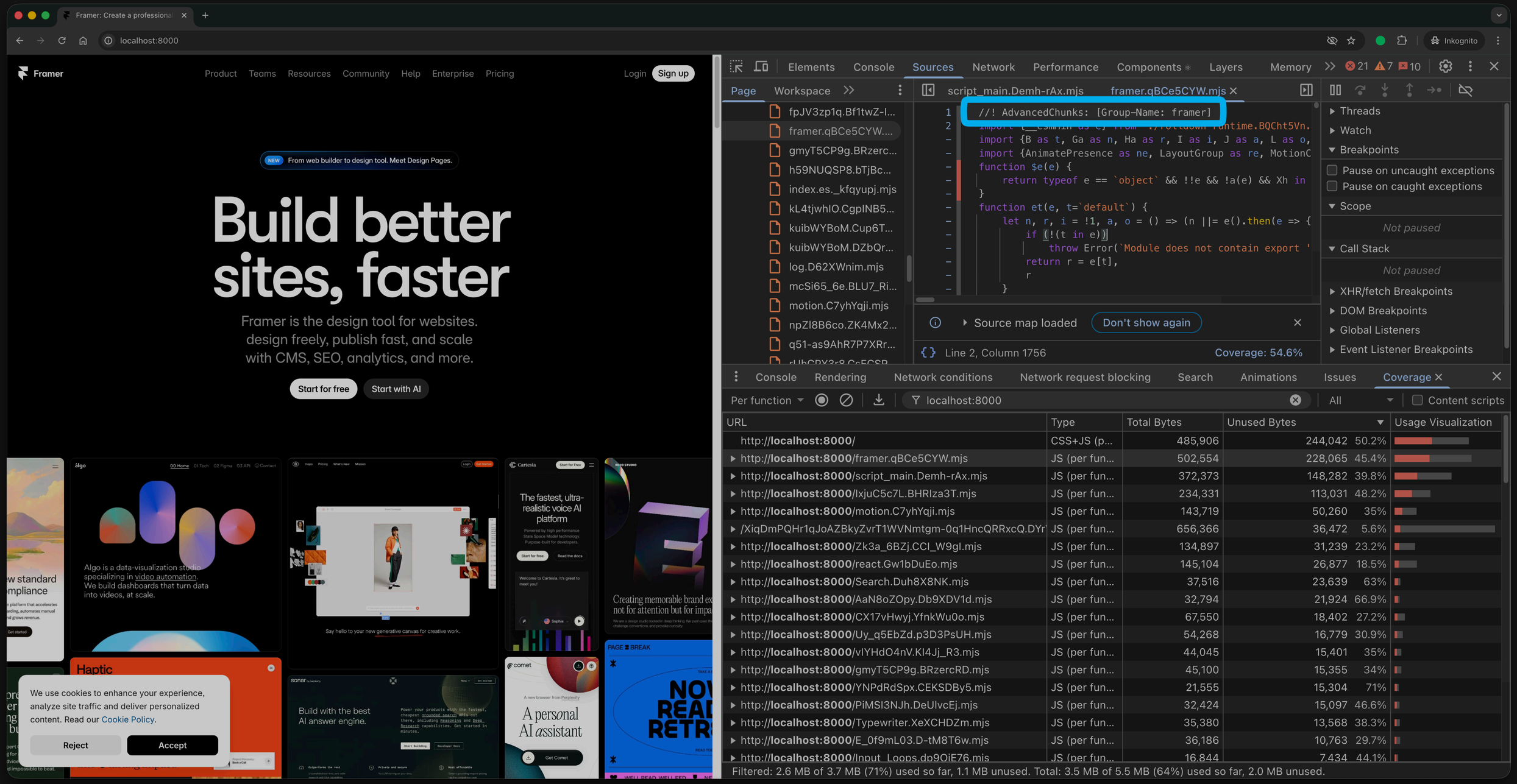
Analyzing bundles

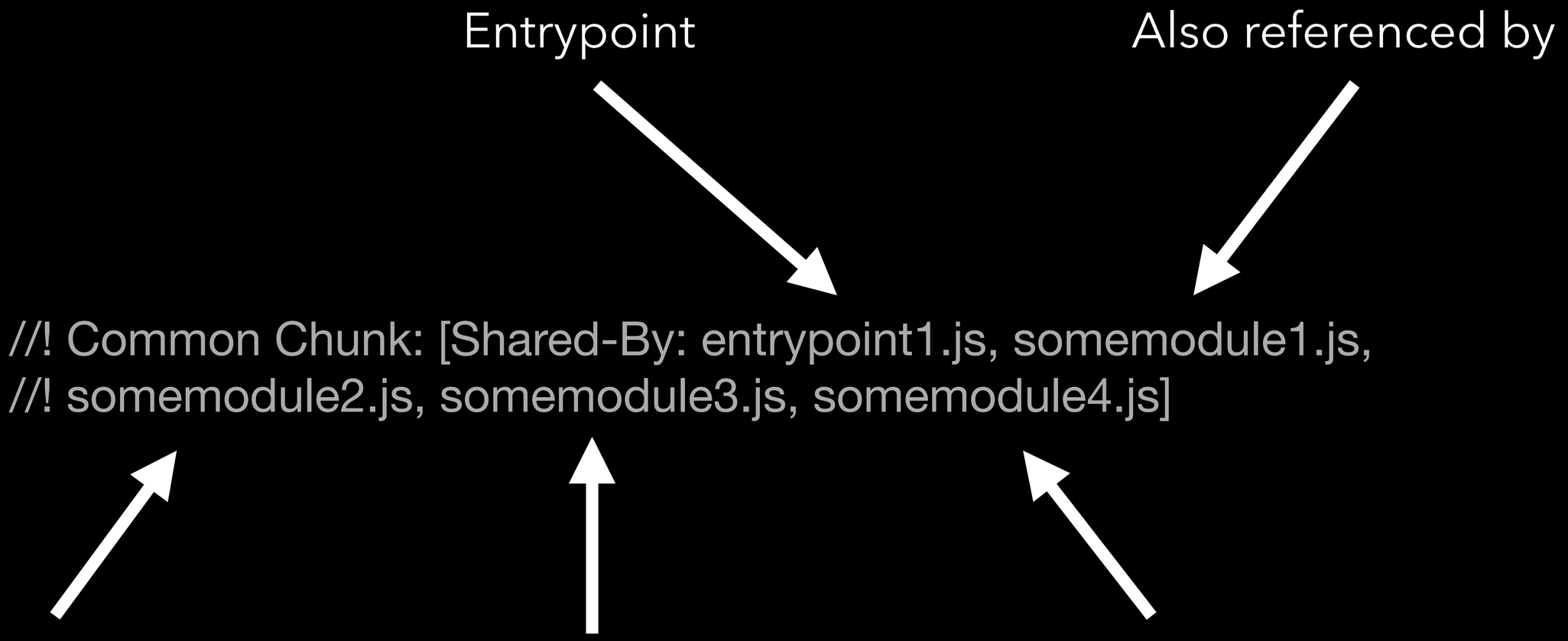
Follow discussions about emitting an esbuild-like metafile in #6425.

Once we have that, you can use:

- hawkeye, bundle-buddy, esbuild-analyzer, Bundle Size Analyzer
- @rnx-kit/esbuild-bundle-analyzer
  …which unlocks support for microsoft/webpack-bundle-compare &  Rsdoctor

```
experimental: {
    attachDebugInfo: "full",
},
```


Framer: Create a professional
localhost:8000
Inkognito
Framer
Product
Teams
Resources
Community
Help
Enterprise
Pricing
Login
Sign up
NEW From web builder to design tool. Meet Design Pages.
Build better sites, faster
Framer is the design tool for websites. design freely, publish fast, and scale with CMS, SEO, analytics, and more.
Start for free
Start with AI
Algo is a data-visualization studio specializing in video automation. We build dashboards that turn data into videos, at scale.
Say hello to your new generative canvas for creative work.
The fastest, ultra-realistic voice AI platform
Haptic
We use cookies to enhance your experience, analyze site traffic and deliver personalized content. Read our Cookie Policy.
Reject
Accept
Build with the best AI answer engine.
A personal AI assistant
Get Comet
Elements
Console
Sources
Network
Performance
Components
Layers
Memory
Page
Workspace
script_main.Demh-rAx.mjs
framer.qBCe5CYW.mjs
fpJV3zp1q.Bf1twZ-I...
framer.qBCe5CYW....
gmyT5CP9g.BRzerc...
h59NUQSP8.bTjBc...
index.es._kfqyupj.mjs
kL4tjwhIO.CgpINB5...
kuibWYBoM.Cup6T...
kuibWYBoM.DZbQr...
log.D62XWnim.mjs
mcSi65_6e.BLU7_Ri...
motion.C7yhYqji.mjs
npZl8B6co.ZK4Mx2...
q51-as9AhR7P7XRr...
//! AdvancedChunks: [Group-Name: framer]
import {B as t, Ga as n, Ha as r, I as i, J as a, L as o,
import {AnimatePresence as ne, LayoutGroup as re, MotionC
function $e(e) {
return typeof e == `object` && !!e && !a(e) && Xh in
}
function et(e, t=`default`) {
let n, r, i = !1, a, o = () => (n ||= e().then(e => {
if (!(t in e))
throw Error(`Module does not contain export '
return r = e[t],
r
}
Source map loaded
Don't show again
Line 2, Column 1756
Coverage: 54.6%
Threads
Watch
Breakpoints
Pause on uncaught exceptions
Pause on caught exceptions
Scope
Not paused
Call Stack
Not paused
XHR/fetch Breakpoints
DOM Breakpoints
Global Listeners
Event Listener Breakpoints
Console
Rendering
Network conditions
Network request blocking
Search
Animations
Issues
Coverage
Per function
localhost:8000
All
Content scripts
URL
Type
Total Bytes
Unused Bytes
Usage Visualization
http://localhost:8000/ CSS+JS (p... 485,906 244,042 50.2%
http://localhost:8000/framer.qBCe5CYW.mjs JS (per fun... 502,554 228,065 45.4%
http://localhost:8000/script_main.Demh-rAx.mjs JS (per fun... 372,373 148,282 39.8%
http://localhost:8000/lxjuC5c7L.BHRIza3T.mjs JS (per fun... 234,331 113,031 48.2%
http://localhost:8000/motion.C7yhYqji.mjs JS (per fun... 143,719 50,260 35%
/XiqDmPQHr1qJoAZBkyZvrT1WVNmtgm-0q1HncQRRxcQ.DYr JS (per fun... 656,366 36,472 5.6%
http://localhost:8000/Zk3a_6BZj.CCl_W9gI.mjs JS (per fun... 134,897 31,239 23.2%
http://localhost:8000/react.Gw1bDuEo.mjs JS (per fun... 145,104 26,877 18.5%
http://localhost:8000/Search.Duh8X8NK.mjs JS (per fun... 37,516 23,639 63%
http://localhost:8000/AaN8oZOpy.Db9XDV1d.mjs JS (per fun... 32,794 21,924 66.9%
http://localhost:8000/CX17vHwyj.YfnkWu0o.mjs JS (per fun... 67,550 18,402 27.2%
http://localhost:8000/Uy_q5EbZd.p3D3PsUH.mjs JS (per fun... 54,268 16,779 30.9%
http://localhost:8000/vIYHdO4nV.KI4Jj_R3.mjs JS (per fun... 44,045 15,401 35%
http://localhost:8000/gmyT5CP9g.BRzercRD.mjs JS (per fun... 45,100 15,355 34%
http://localhost:8000/YNPdRdSpx.CEKSDBy5.mjs JS (per fun... 21,555 15,304 71%
http://localhost:8000/PiMSI3NJh.DeUlvcEj.mjs JS (per fun... 32,424 15,097 46.6%
http://localhost:8000/Typewriter.XeXCHDZm.mjs JS (per fun... 35,380 13,568 38.3%
http://localhost:8000/E_0f9mL03.D-tM8T6w.mjs JS (per fun... 36,186 10,763 29.7%
Filtered: 2.6 MB of 3.7 MB (71%) used so far, 1.1 MB unused. Total: 3.5 MB of 5.5 MB (64%) used so far, 2.0 MB unused.



To get rid of this chunk → make sure it’s not imported by those modules

Results

Improvements to chunks

p25	20 → **14**
p75	67 → **22**
p90	80 → **30**
p99	95 → **54**

36%

Median drop in download size in both un- and compressed JS

4%

Avg. improvement to LCP[1] for websites with 1–2MB of JS

[1] LCP adjusted to TTFB

41%

Avg. improvement to LCP[1] for websites with 2MB+ of JS.

[1] LCP adjusted to TTFB

11%

Faster LCP[1] at p90 across all Framer sites.
❤️ for slow devices & slow networks

[1] LCP adjusted to TTFB

A fast, stable and capable bundler is a win-win for everyone.

For us, our customers & their visitors, the Vite community, the web.

Thank you!

Questions?

@kurtextrem on X, LinkedIn, BlueSky

https://kurtextrem.de