

ANGULAR
ARCHITECTS

Performance Audit Tools

Alexander Thalhammer | @LX_T



Img src: <https://bit.ly/ng-tools-img>

Performance Audit Tools

- Lighthouse (PageSpeed.com) or
- WebPageTest.org
- Source Map Explorer or
- Webpack Bundle Analyzer
- Import Graph Visualizer
- Chrome DevTools
- Angular DevTools CD Profiler

Performance Audit Tools

- Lighthouse (PageSpeed.com)
 - WebPageTest.org
 - Source Map Explorer
 - Webpack Bundle Analyzer
 - Import Graph Visualizer
 - Chrome DevTools
 - Angular DevTools CD Profiler
-
- The diagram illustrates the categorization of performance audit tools. A vertical list of eight tools is presented, with three pink curly braces on the right side grouping them into three categories: 'Initial Load', 'Bundle Size', and 'Runtime'. The 'Initial Load' group contains the first two tools. The 'Bundle Size' group contains the next three tools. The 'Runtime' group contains the last three tools.
- Lighthouse (PageSpeed.com)
 - WebPageTest.org
 - Source Map Explorer
 - Webpack Bundle Analyzer
 - Import Graph Visualizer
 - Chrome DevTools
 - Angular DevTools CD Profiler
- Initial Load
- Bundle Size
- Runtime

Web Audit Tools



PageSpeed Insights vs Chrome Lighthouse

- PageSpeed Insights
- **Real user data & Lighthouse lab results**
- Go to pagespeed.web.dev & enter URL
- Test is being run on Google servers
- Performance, new since 2023
 - Accessibility
 - Best Practices
 - SEO
- Chrome Lighthouse
- Open URL in your browser
(run in incognito mode, close other Apps)
- Open DevTools Lighthouse tab → Analyze
- Performance, other tests
 - Accessibility
 - Best Practices
 - SEO
 - **PWA**



Demo

PageSpeed Insights

PageSpeed Score – Case Studies

Good examples

- Websites
 - [WP Rocket](#)
 - [Wikipedia](#)
- Angular Apps
 - <https://www.emilfrey.ch/>

Chrome Lighthouse – Getting Started

- Lighthouse is tool to test NG App **during development**
- We still need a **production build** to get useful results
- How can we run a production build on localhost?

Chrome Lighthouse – How To

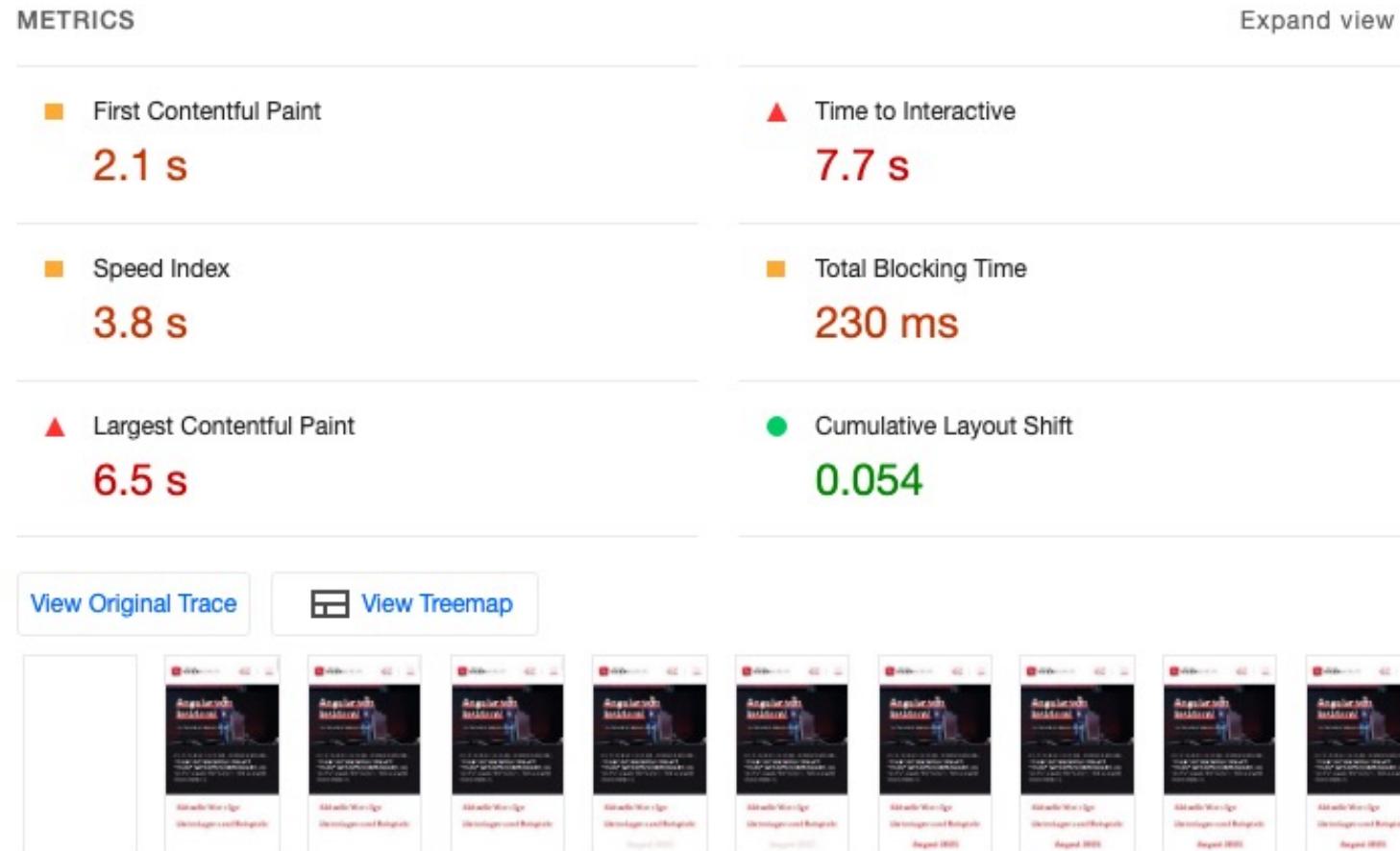
1. Prod build → `ng b(uild)`
2. Serve the build on your localhost with
 - Using localhost tool like MAMP / WAMP / XAMPP
 - NPM serve
 - <https://www.npmjs.com/package/serve>
 - Or edit hosts file manually if you know how to do that ☺
3. Open in Chrome and run Lighthouse



Demo

Chrome Lighthouse

Chrome Lighthouse – Details



Important Metrics

What's being measured?

- Time to First Byte (TTFB)
- First Contentful Paint (FCP)
- Speed Index (originally by WebPageTest)
- Largest Contentful Paint (LCP)
- Time to Interactive (TTI)
- Total Blocking Time (TBT) → TTI - FCP
- Cumulative Layout Shift (CLS)
- First Input Delay (FID)
- Interaction to Next Paint (INP)

WebPageTest.org

- An alternative to Google's PageSpeed & Lighthouse
- Measures same things (TTFB, FCP, Speed Index, LCP, TBT, CLS)
- Generates waterfall, screenshot, video & a content breakdown
- Runs multiple tests at once (e.g. 3 or 5) and select median
- Choose test location and configure test machine
- Lots of further advanced settings, like HTTP Authentication

WebPageTest – Summary

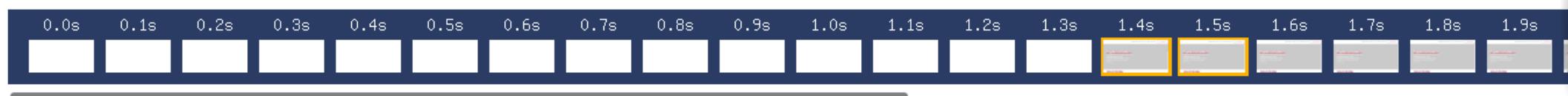
Observed Metrics (Based on Median Run by: ▶ Speed Index)

ⓘ Note: Metrics offered will vary.

FIRST VIEW (RUN 1)

First Byte	Start Render	FCP	Speed Index	LCP	CLS	TBT	Total Bytes
.479s	1.400s	1.370s	3.112s	3.307s	.002	.482s	1,976 KB

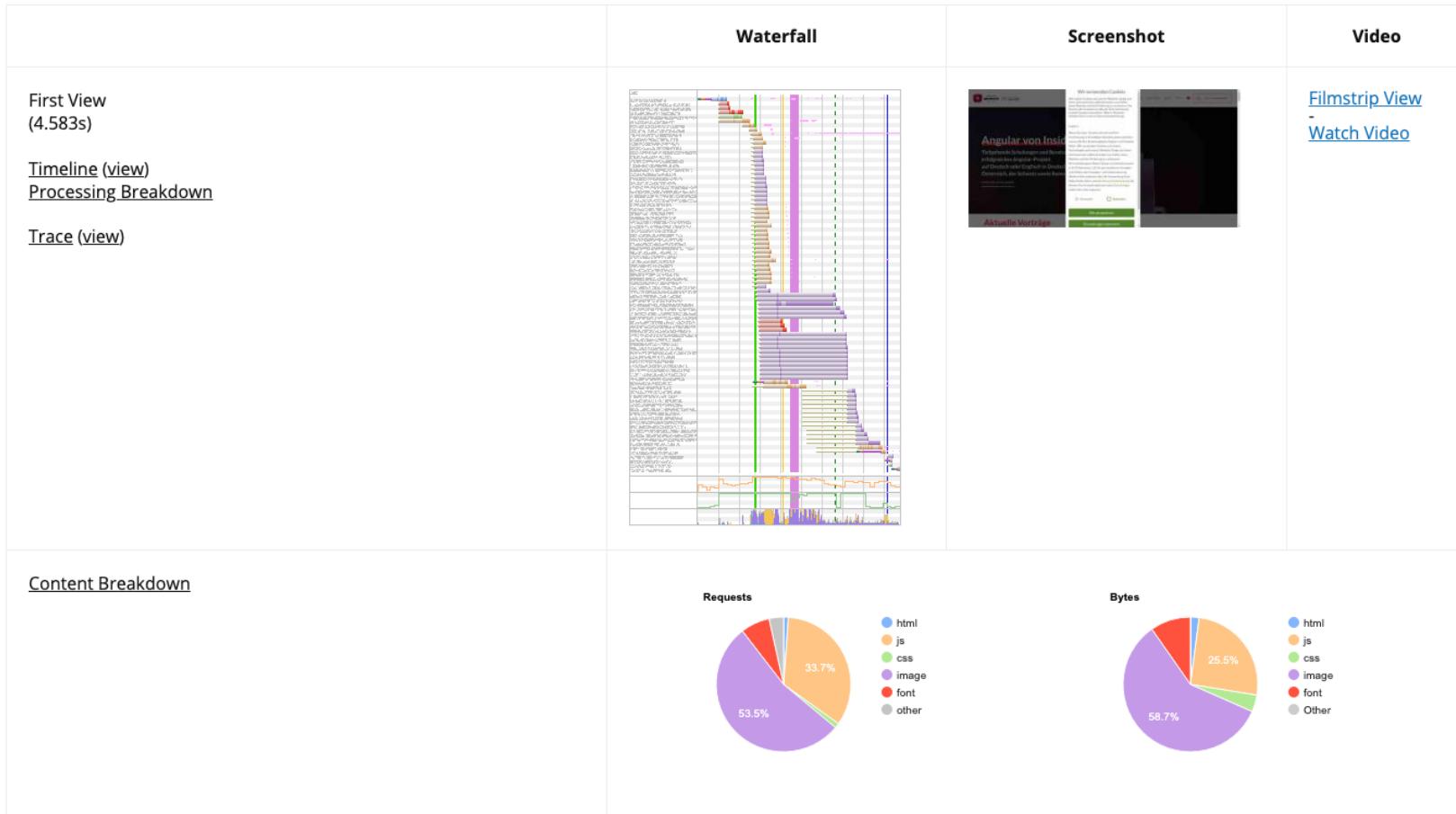
Visual Page Loading Process (Explore)



[Compare First Views](#)

[Plot Full Results](#)

WebPageTest – Details





Demo

WebPageTest

Perfume.js

- Tiny, web performance monitoring library
 - <https://github.com/Zizzamia/perfume.js/>
- Reports field data back to analytics tool
 -  Supports latest Performance APIs for precise metrics
 -  Device data enrichment
 -  Cross browser tested
 -  Filters out false positive/negative results
 -  Only 5.1Kb gzip
 -  Web Vitals Score
 -  Flexible analytics tool
 -  Waste-zero ms with [requestIdleCallback](#) strategy built-in

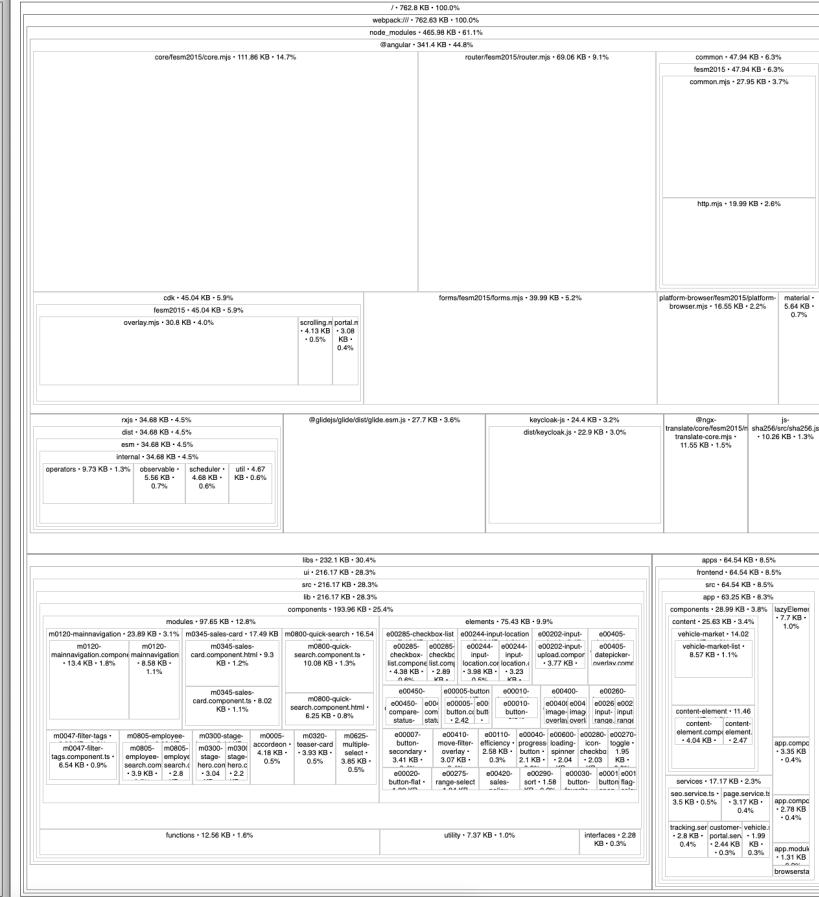
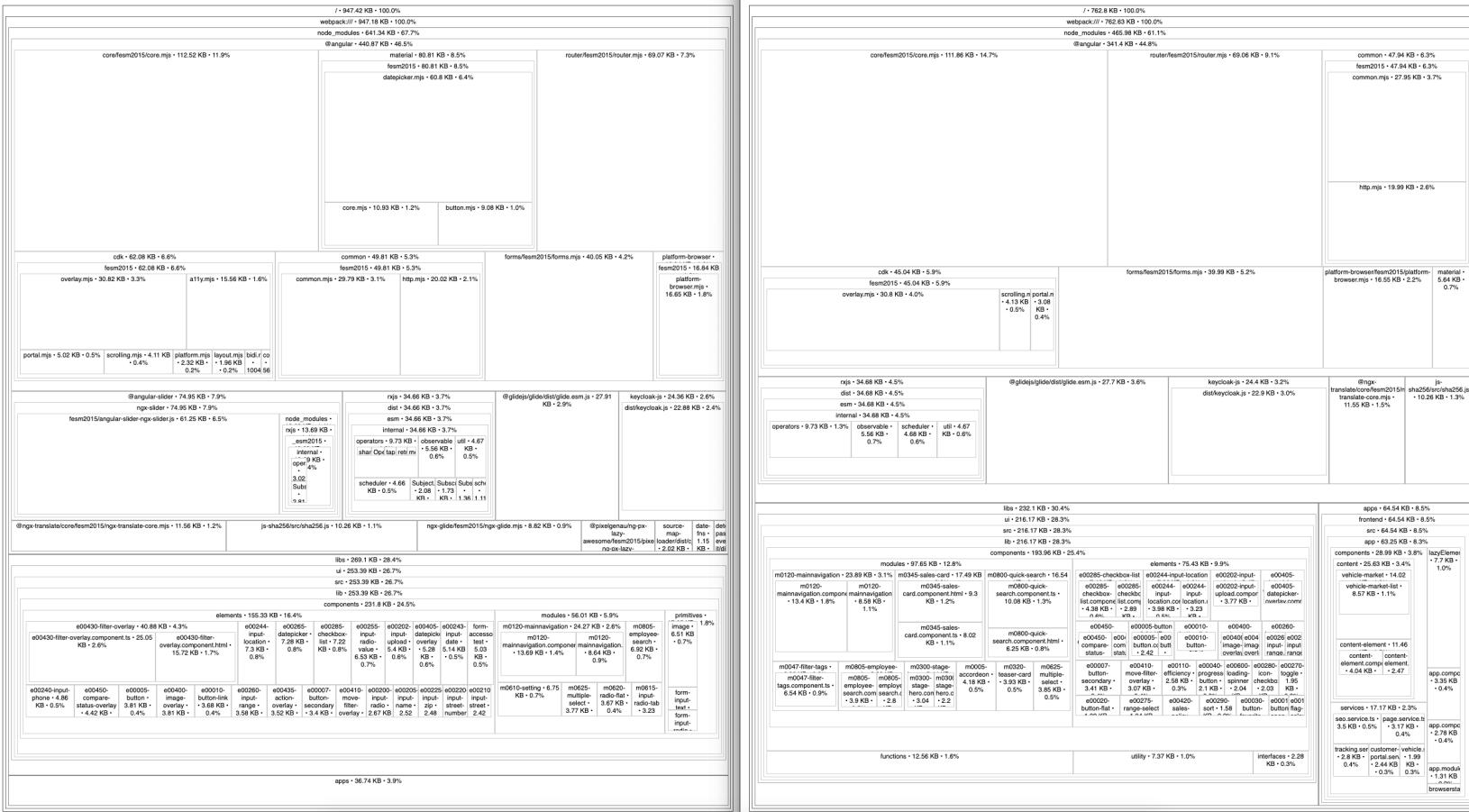
Build Analysis Tools



Source Map Explorer

- Needs generated source maps to work
 - Works for esbuild and webpack
 - Either set in build options (angular.json) or use build flag "--source-map"
- Analyzes a single js file or whole bundle / build
 - main bundle
 - or lazy loading bundles
- Determines where each byte in your code comes from
- Shows a black/white treemap visualization of a build (size)

Source Map Explorer





Demo

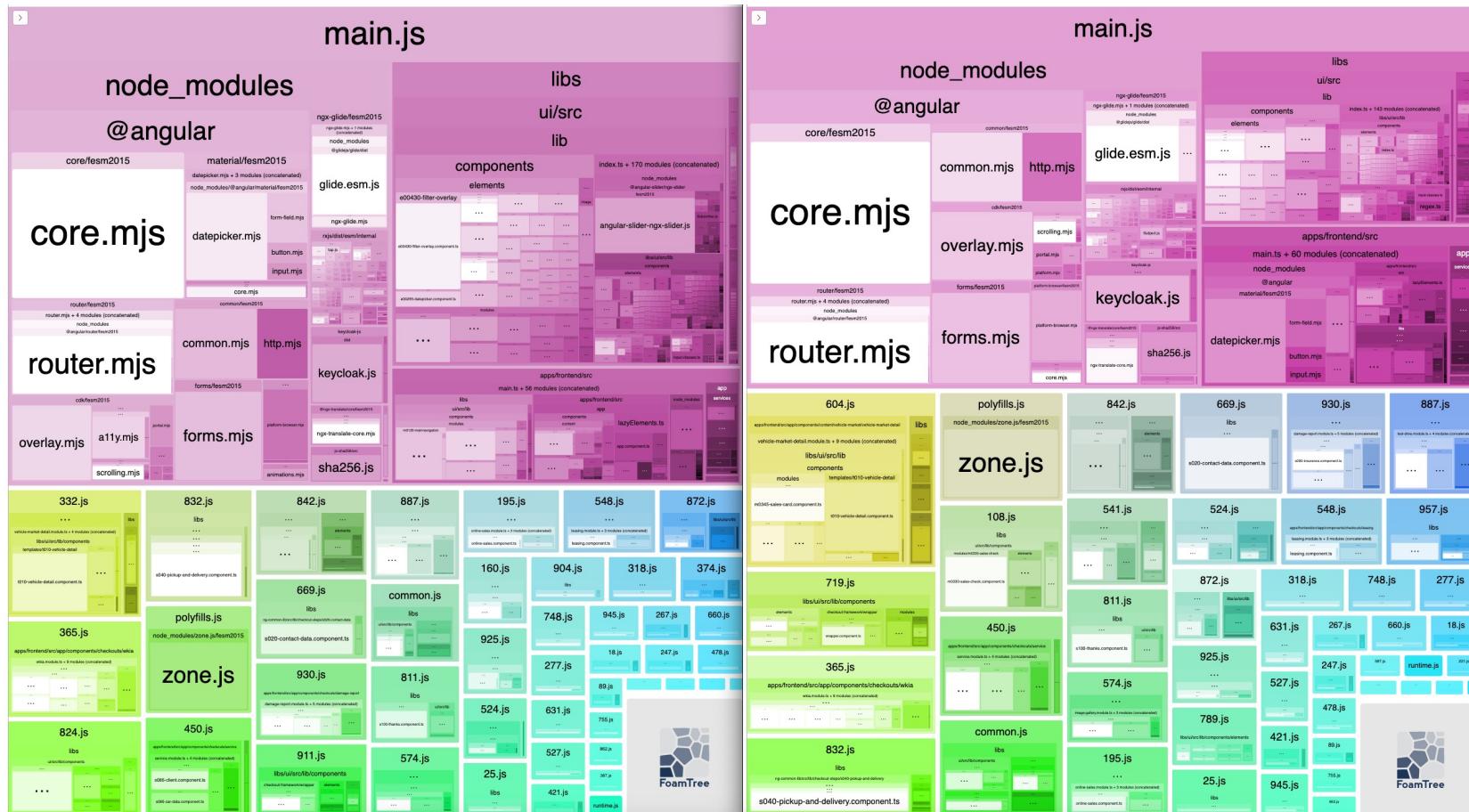
Source Map Explorer

<https://www.npmjs.com/package/source-map-explorer>

Webpack Bundle Analyzer (webpack only!)

- Needs webpack and a generated stats.json to work
 - Needs "builder": "@angular-devkit/build-angular:browser"
 - Either set in build options (angular.json) or build flag "--stats-json"
- Analyzes the whole build
- Visualize size of all js chunks
 - Good to analyze lazy loading
- Interactive, zoomable and colorful treemap ☺

Webpack Bundle Analyzer



ANGULAR ARCHITECTS



Demo

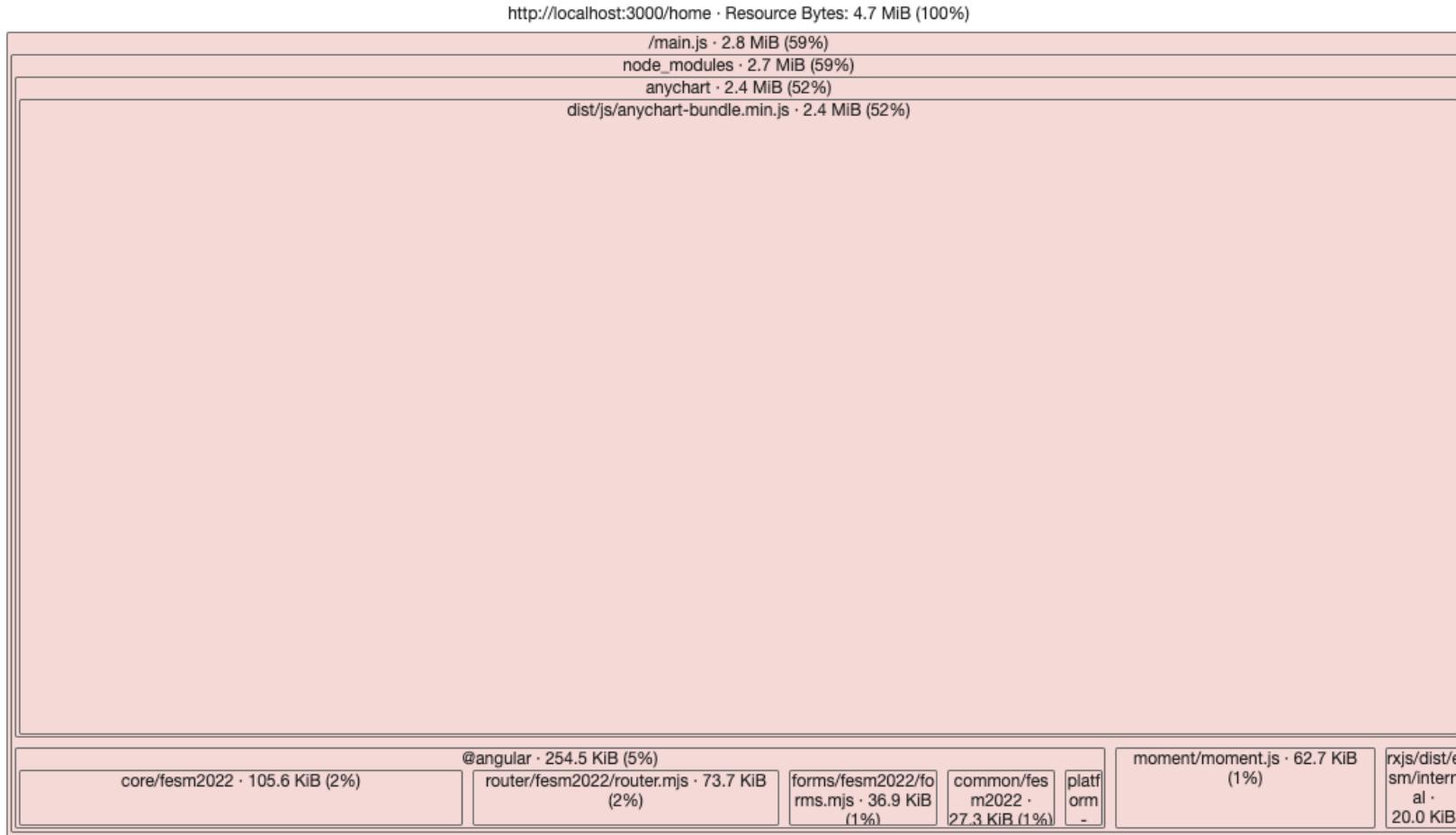
Webpack Bundle Analyzer

<https://github.com/webpack-contrib/webpack-bundle-analyzer>

Lighthouse Treemap

- Also similar to Source Map Explorer
- Also analyzes the whole build
- But generated by Chrome – no extra build needed ☺

Lighthouse Treemap





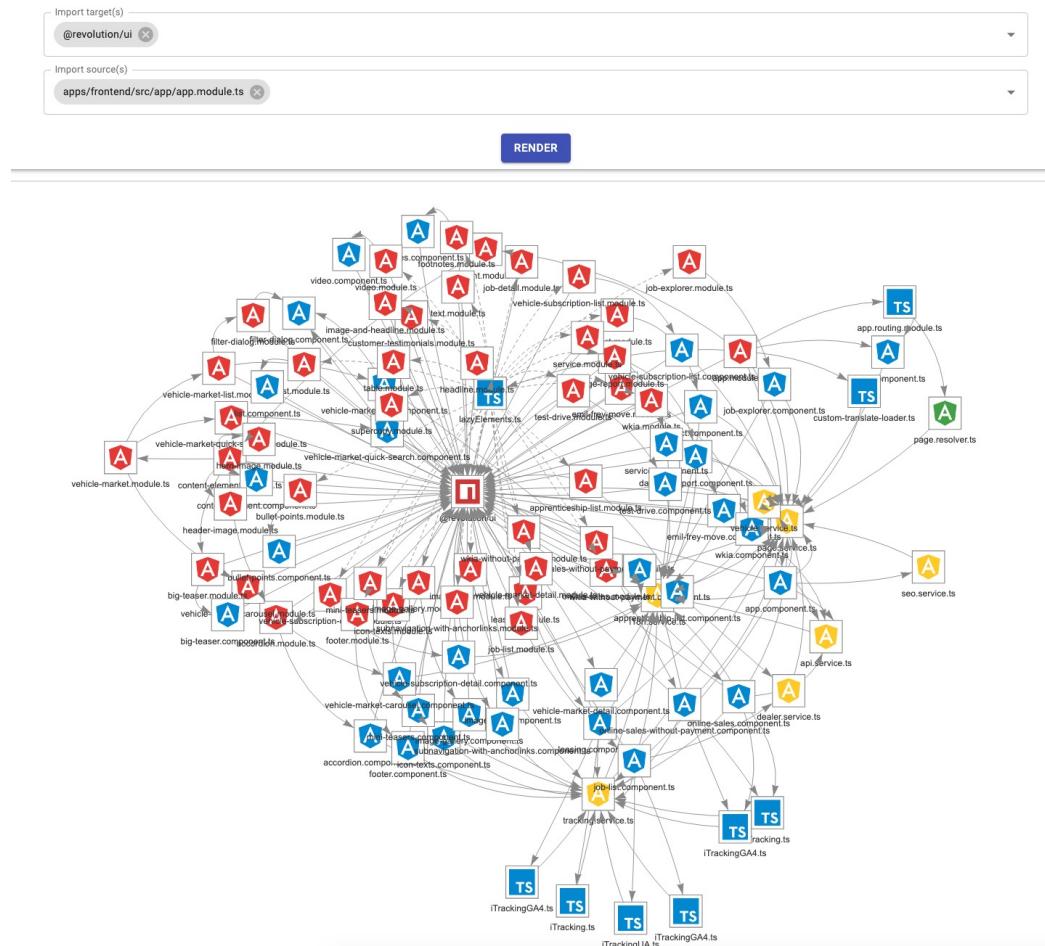
Demo

Lighthouse Treemap

Import Graph Visualizer

- A development tool for filtering and visualizing import paths within a JavaScript/TypeScript application
- Allows filtering import paths by source & target modules
- Allows zooming in to a limited subsection of your app, which will likely be easier to analyze than the entire app

Import Graph Visualizer



Import Graph Visualizer – How To

- `npx import-graph-visualizer --entry-points path/to/entry/module --ts-config path/to/tsconfig`
- e.g. `npx import-graph-visualizer --entry-points src/main.ts --ts-config tsconfig.app.json`



Demo

Import Graph Visualizer

<https://github.com/rx-angular/import-graph-visualizer>

Lab 01 Performance Audits

Lighthouse / Build Analysis

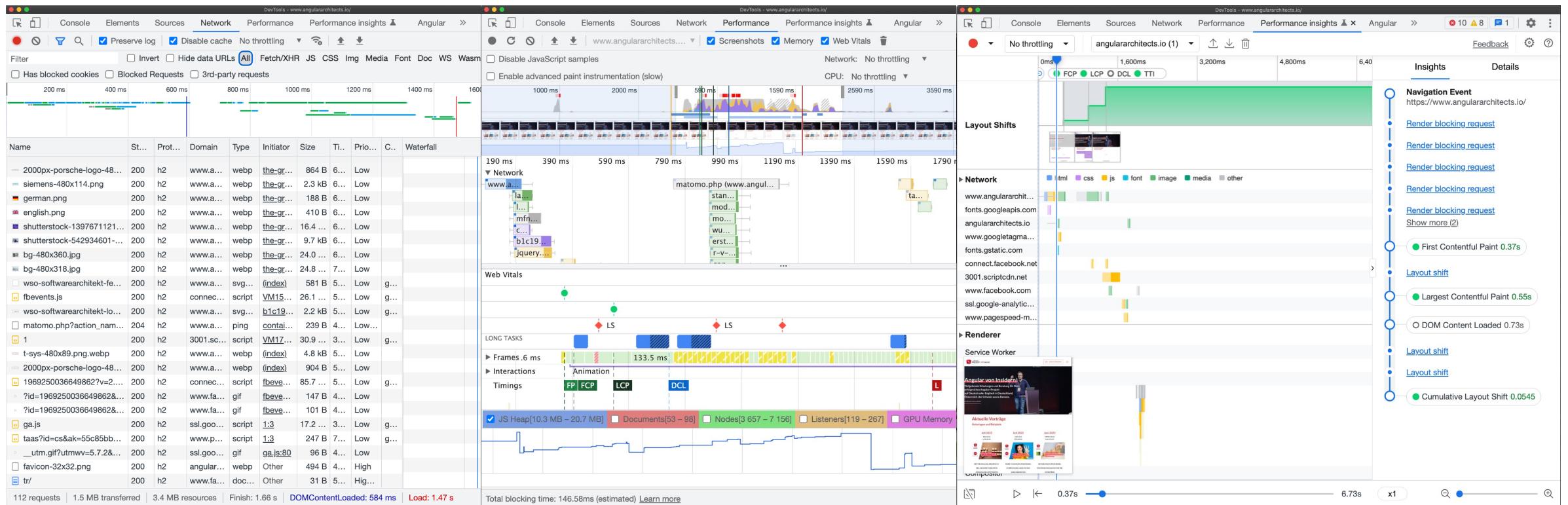
Runtime Audit Tools



Google Chrome DevTools

- The Chrome DevTools are not only used for
 - Styling (Elements)
 - Debugging (Console)
- But also for Performance
 - Network
 - Performance
 - Memory
 - memory heap comparison
 - Performance Insights

Google Chrome DevTools





Demo

Chrome DevTools

Angular Performance Profiling (APP)

- Bridging the Angular framework with Performance panel
- Angular hooks into DevTools Performance panel
- adds an **“Angular” track**
- logs framework-specific events like
 - change detection
 - lifecycle hooks and
 - DI operations

APP – enable profiling

- Running `ng.enableProfiling()` in the Chrome console or
- Calling `enableProfiling()` before `bootstrapApplication()`
- important: DevTools work only in DevMode

APP – start recording

- DevTools → Performance tab
- Hit **Record**, interact with app (e.g. reload), then **Stop**
- DevTools captures both browser-level and Angular-specific metrics together

APP – interprete profile

Color-coded flame chart

- **Blue** → your TS methods (constructors, services, etc.)
- **Purple** → compiled template activity
- **Green** → entry points like lifecycle hooks or CD triggers

APP – helps spotting

- Slow bootstraps (e.g. DI and initial CD in green/blue)
- Heavy or redundant CD flows
- Unnecessary multiple CD runs
- Template or lifecycle hooks causing performance issues



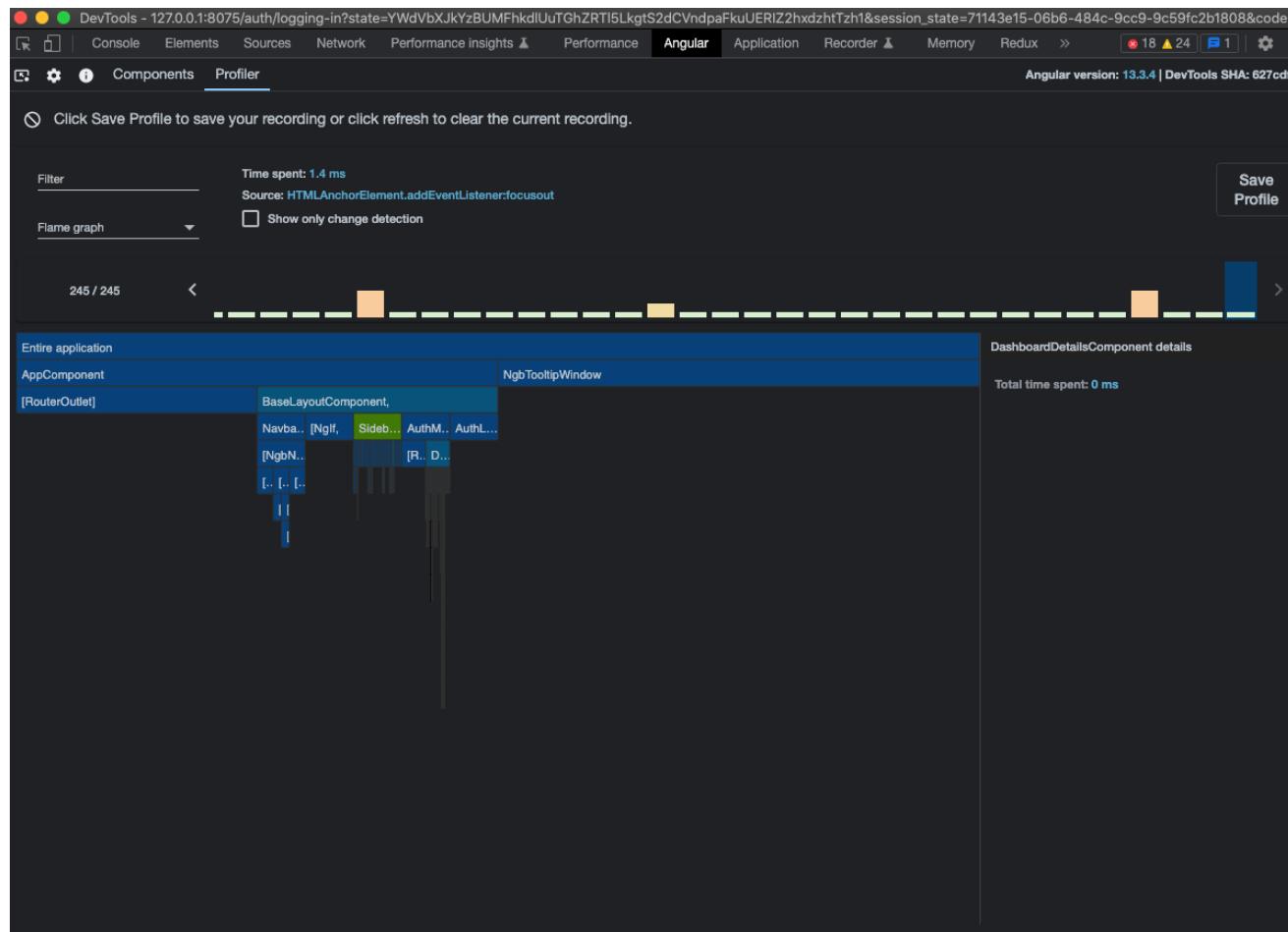
Demo

Chrome DevTools

Angular DevTools Profiler

- Angular DevTools extension can be added to Chrome
 - <https://chromewebstore.google.com/detail/angular-devtools/ienfalfjdbdpebioblackkekamfmbnh>
 - Features a **Component Tree** to inspect the components
 - **Profiler**
 - And the newly added **Injector Tree** (Dependency Injection)
- Profiler shows individual change detection (CD) cycles
 - What triggered CD
 - How much time it took executing CD

Angular DevTools Profiler





Demo

Angular DevTools Profiler

Performance Audit Tools

- Lighthouse (PageSpeed.com)
 - WebPageTest.org
 - Source Map Explorer
 - Webpack Bundle Analyzer
 - Import Graph Visualizer
 - Chrome DevTools
 - Angular DevTools CD Profiler
-
- The diagram illustrates the categorization of performance audit tools. It features three groups of tools, each enclosed in a pink curly brace. The first group, labeled 'Initial Load', contains Lighthouse and WebPageTest.org. The second group, labeled 'Bundle Size', contains Source Map Explorer, Webpack Bundle Analyzer, and Import Graph Visualizer. The third group, labeled 'Runtime', contains Chrome DevTools and Angular DevTools CD Profiler.

References

- Google Web Dev
 - <https://pagespeed.web.dev>
 - <https://web.dev/metrics/>
- Improving Load Performance - Chrome DevTools 101
 - <https://www.youtube.com/watch?v=5fLW5Q5ODiE>
- How to analyze your JavaScript bundles
 - <https://www.youtube.com/watch?v=MxBCPc7bQvM>

The background of the slide features a photograph of a modern architectural structure, likely a glass and steel building, viewed from a low angle looking up. The sky above is a vibrant red, creating a dramatic contrast with the building's facade.

Questions?