



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Audit Tools Measurement

Hosted by Alex Thalhammer

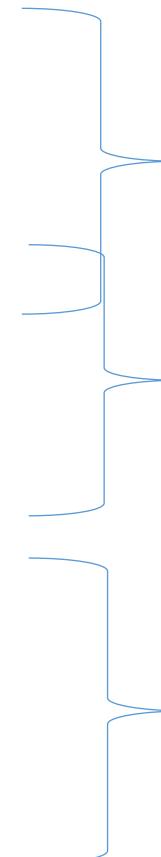


Audit Tools for Measurement

1. Google PageSpeed Insights & Google Chrome Lighthouse
2. WebPageTest
3. Google Chrome DevTools
4. Angular DevTools Profiler
5. Sourcemap Explorer
6. Webpack Bundle Analyzer
7. Import Graph Visualizer

Audit Tools for Measurement

1. Google PageSpeed Insights & Google Chrome Lighthouse
2. WebPageTest
3. Google Chrome DevTools
4. Angular DevTools Profiler
5. Sourcemap Explorer
6. Webpack Bundle Analyzer
7. Import Graph Visualizer



Initial Load

Runtime

Build Size

#1: PageSpeed Insights vs Chrome Lighthouse

PageSpeed Insights

- Real user experience AND Lighthouse lab results
- Go to <https://pagespeed.web.dev> and enter URL
- Test is run on Google servers

Chrome Lighthouse extension

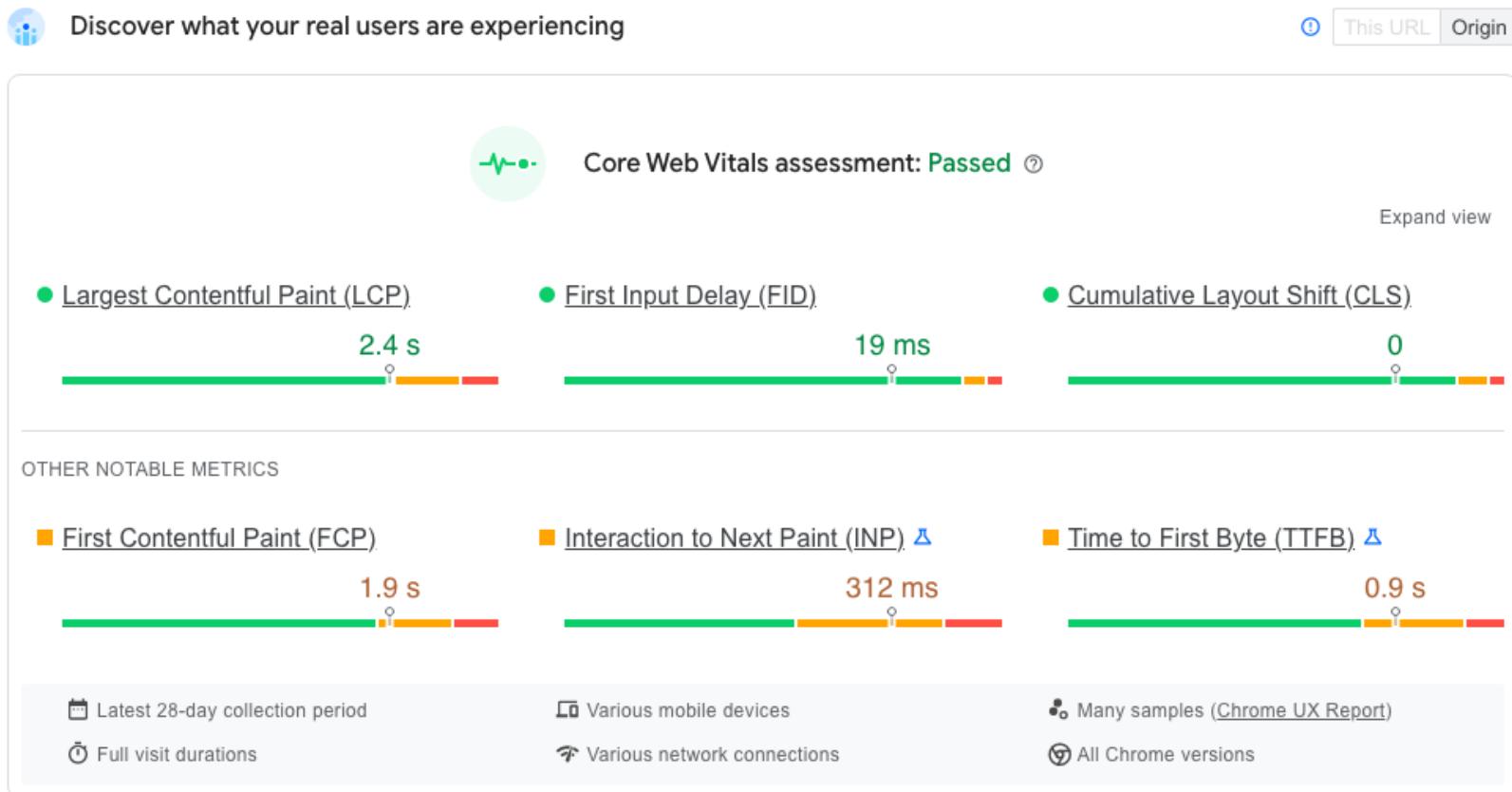
- Lighthouse Performance and other tests
 - Accessibility
 - Best Practices
 - SEO
 - PWA
- Install Lighthouse extension in Chrome
- Open URL and run test on your localhost
(run in incognito mode, close other Apps)

#1: PageSpeed Insights & Chrome Lighthouse

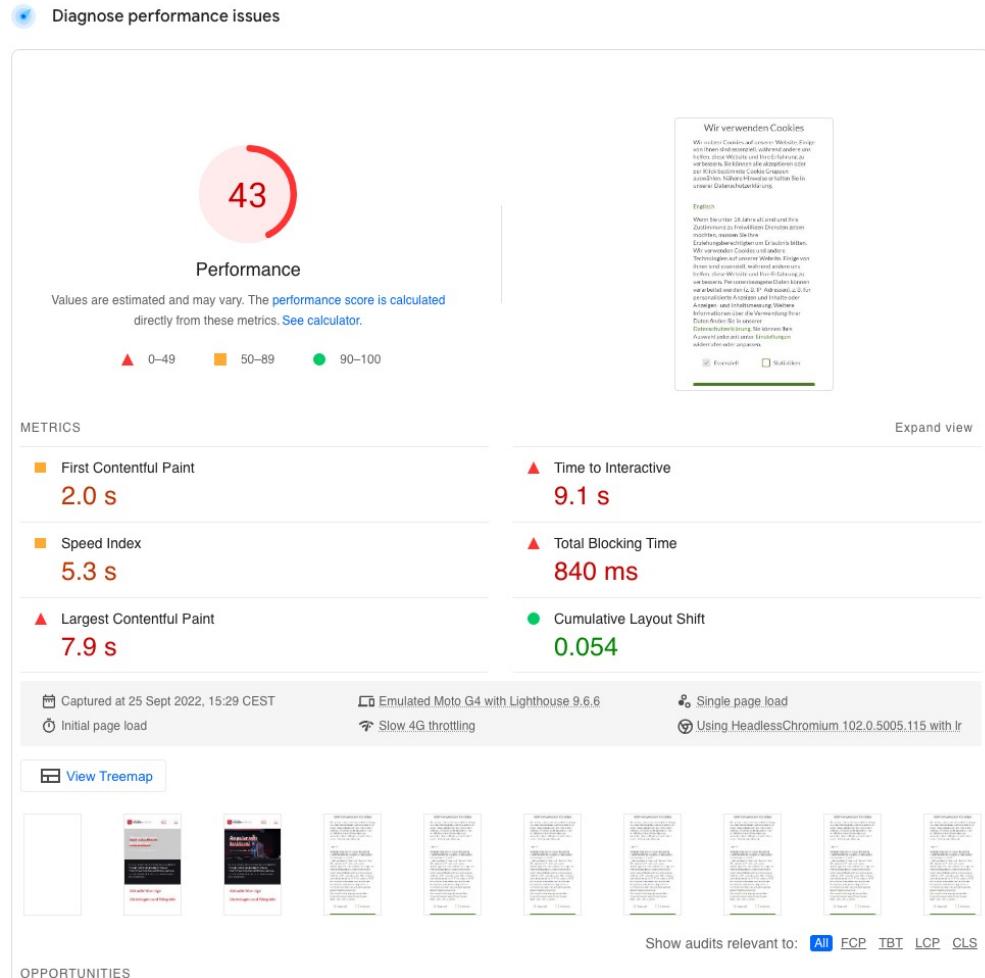
What's being measured?

- 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) → new!

#1: PageSpeed Insights – Real Users (Origin)



#1: PageSpeed Insights – Lab Data (URL)



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



DEMO - PageSpeed Insights

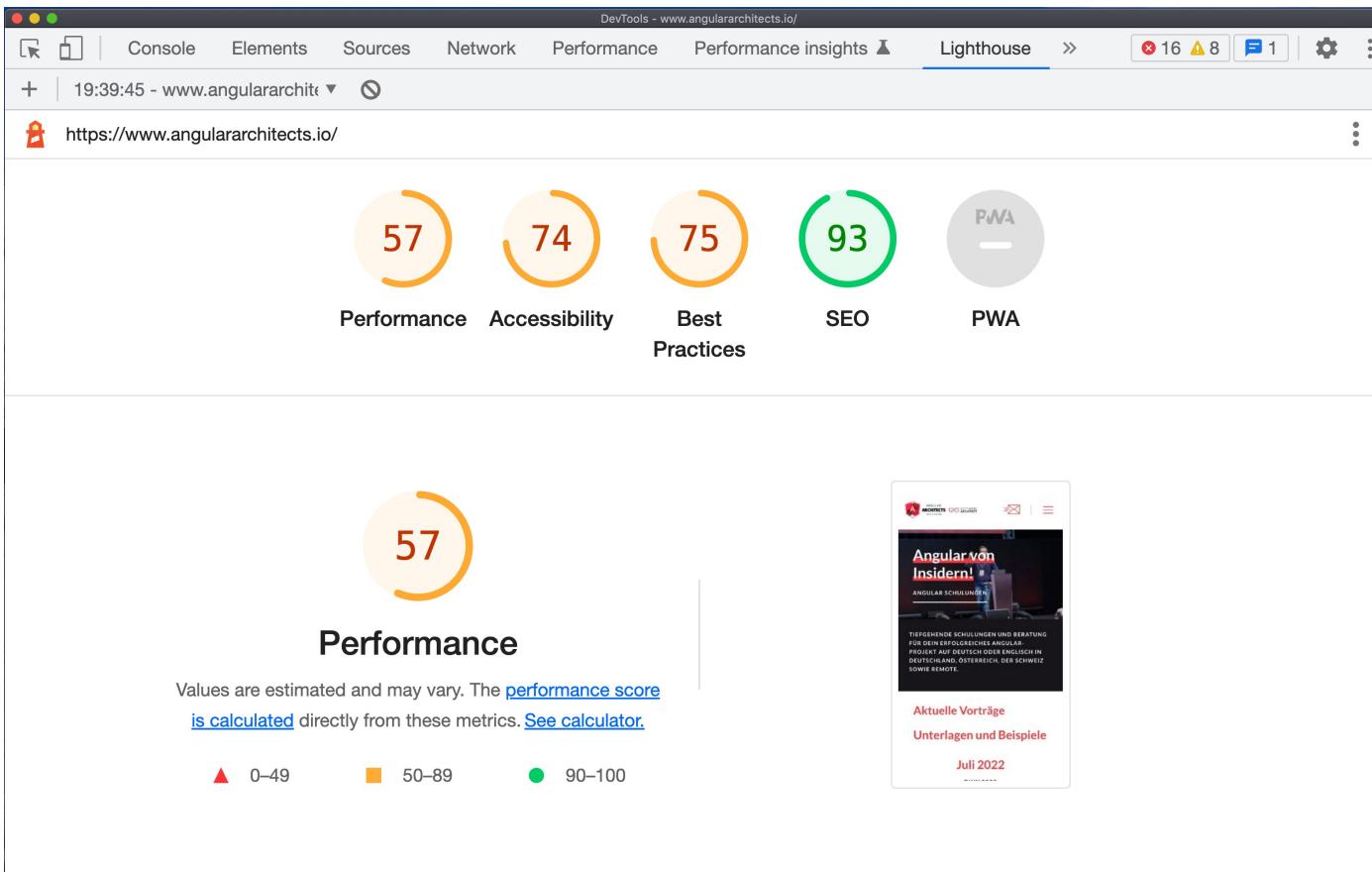
#1: Chrome Lighthouse – Getting Started

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

#1: Chrome Lighthouse – Prod Build Localhost

1. Prod build
 1. ng build --prod
 2. ng build --c production
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

#1: Chrome Lighthouse – Summary

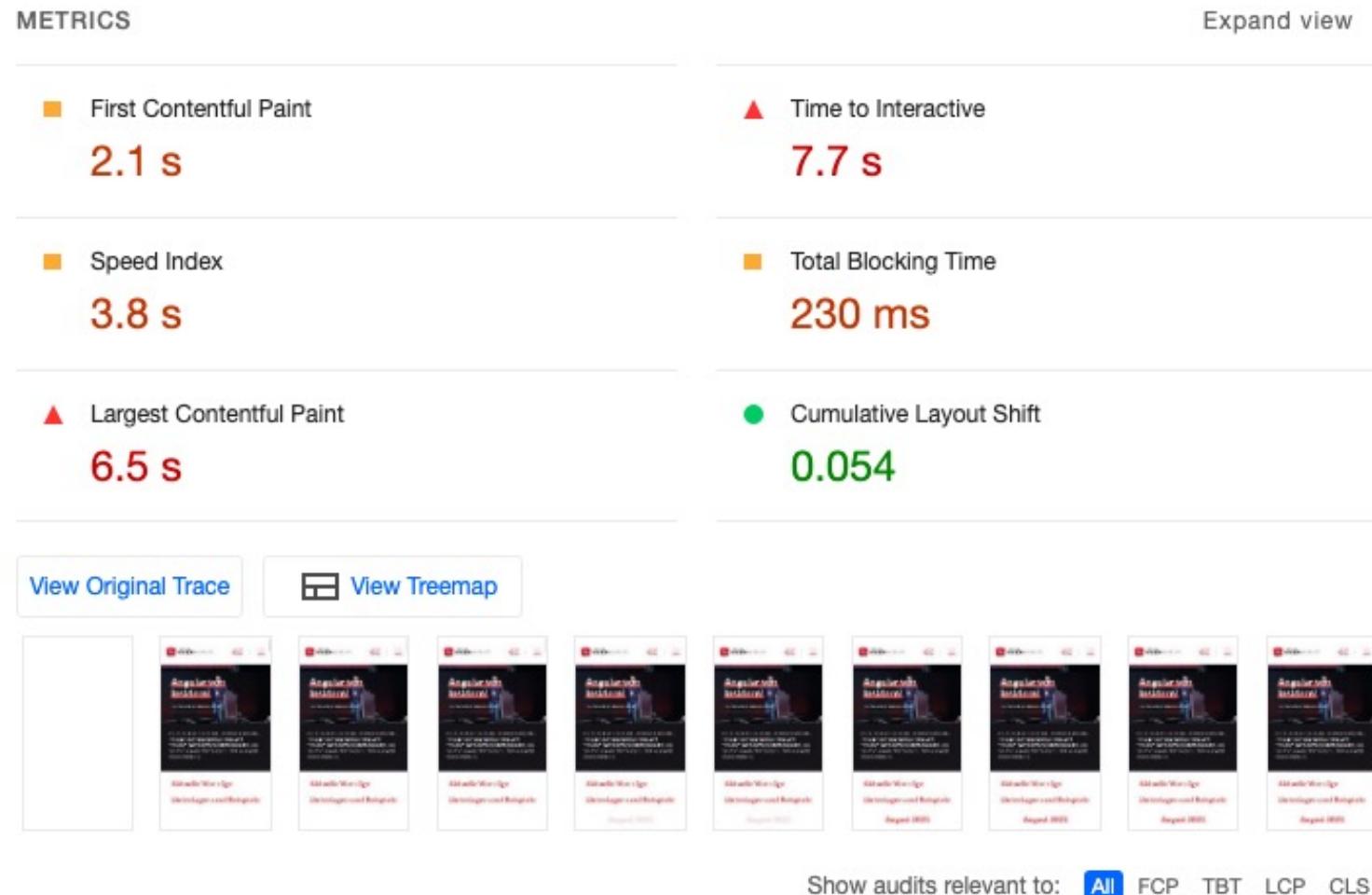


ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

#1: Chrome Lighthouse – Details



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

DEMO - Chrome Lighthouse

#2: WebPageTest

- An alternative to Google's PageSpeed & Lighthouse
- Measures the 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)
- Choose test location and configure test machine
- Lots of further advanced settings

#2: WebPageTest – Summary

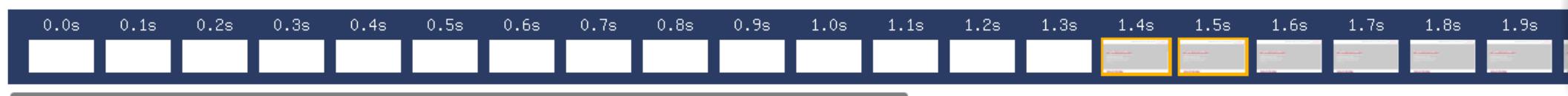
Observed Metrics (Based on Median Run by: ▶ Speed Index)

ⓘ Note: Metrics offered will vary.

FIRST VIEW (RUN 1)

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

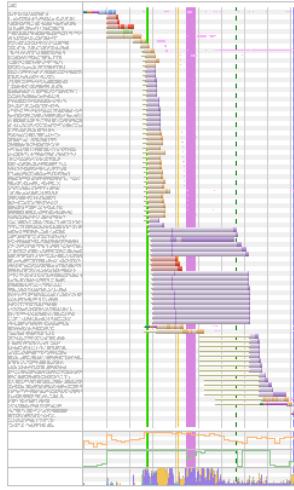
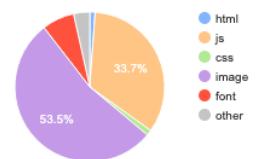
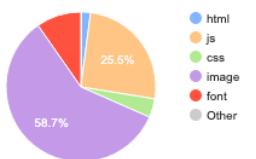
Visual Page Loading Process (Explore)



[Compare First Views](#)

[Plot Full Results](#)

#2: WebPageTest – Details

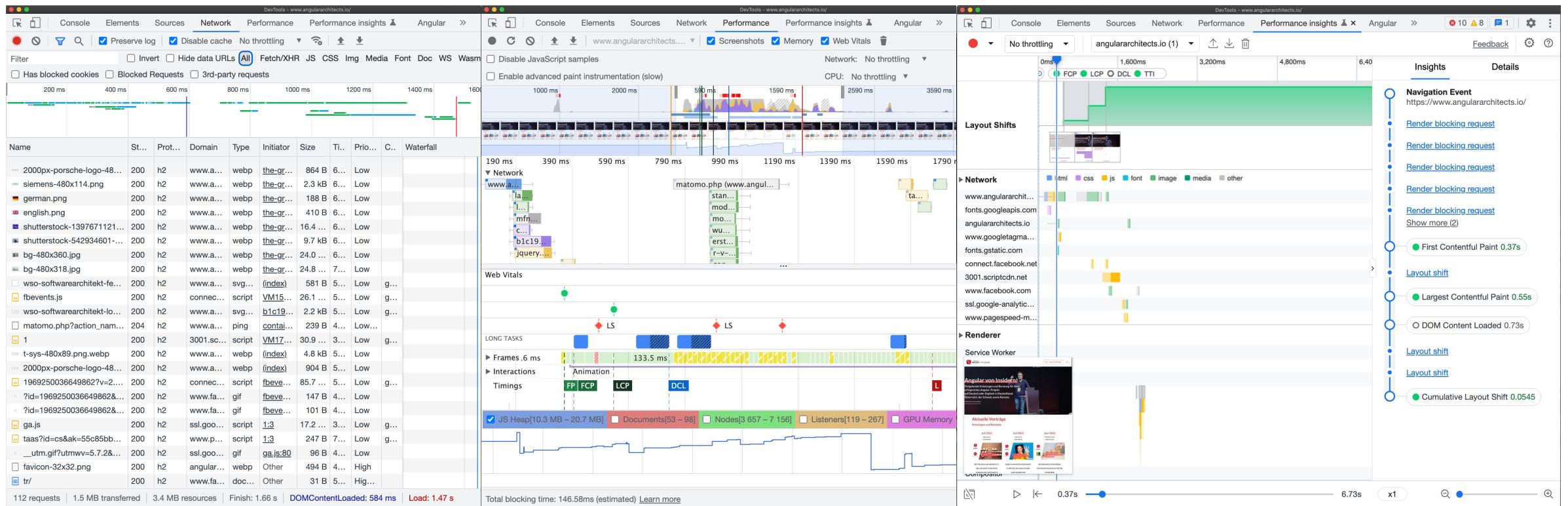
	Waterfall	Screenshot	Video																							
First View (4.583s) Timeline (view) Processing Breakdown Trace (view)			Filmstrip View - Watch Video																							
Content Breakdown	<p>Requests</p>  <table border="1"><thead><tr><th>Type</th><th>Percentage</th></tr></thead><tbody><tr><td>image</td><td>53.5%</td></tr><tr><td>js</td><td>33.7%</td></tr><tr><td>css</td><td>2.1%</td></tr><tr><td>font</td><td>1.8%</td></tr><tr><td>other</td><td>0.7%</td></tr></tbody></table> <p>Bytes</p>  <table border="1"><thead><tr><th>Type</th><th>Percentage</th></tr></thead><tbody><tr><td>image</td><td>58.7%</td></tr><tr><td>js</td><td>25.5%</td></tr><tr><td>css</td><td>2.1%</td></tr><tr><td>font</td><td>1.8%</td></tr><tr><td>other</td><td>0.7%</td></tr></tbody></table>	Type	Percentage	image	53.5%	js	33.7%	css	2.1%	font	1.8%	other	0.7%	Type	Percentage	image	58.7%	js	25.5%	css	2.1%	font	1.8%	other	0.7%	
Type	Percentage																									
image	53.5%																									
js	33.7%																									
css	2.1%																									
font	1.8%																									
other	0.7%																									
Type	Percentage																									
image	58.7%																									
js	25.5%																									
css	2.1%																									
font	1.8%																									
other	0.7%																									

DEMO - WebPageTest

#3: Google Chrome DevTools

- The Chrome DevTools are not only used for
 - Styling (Elements)
 - Debugging (Console)
- But also for Performance
 - Network
 - Performance
 - Performance Insights (Beta!)

#3: Google Chrome DevTools



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE



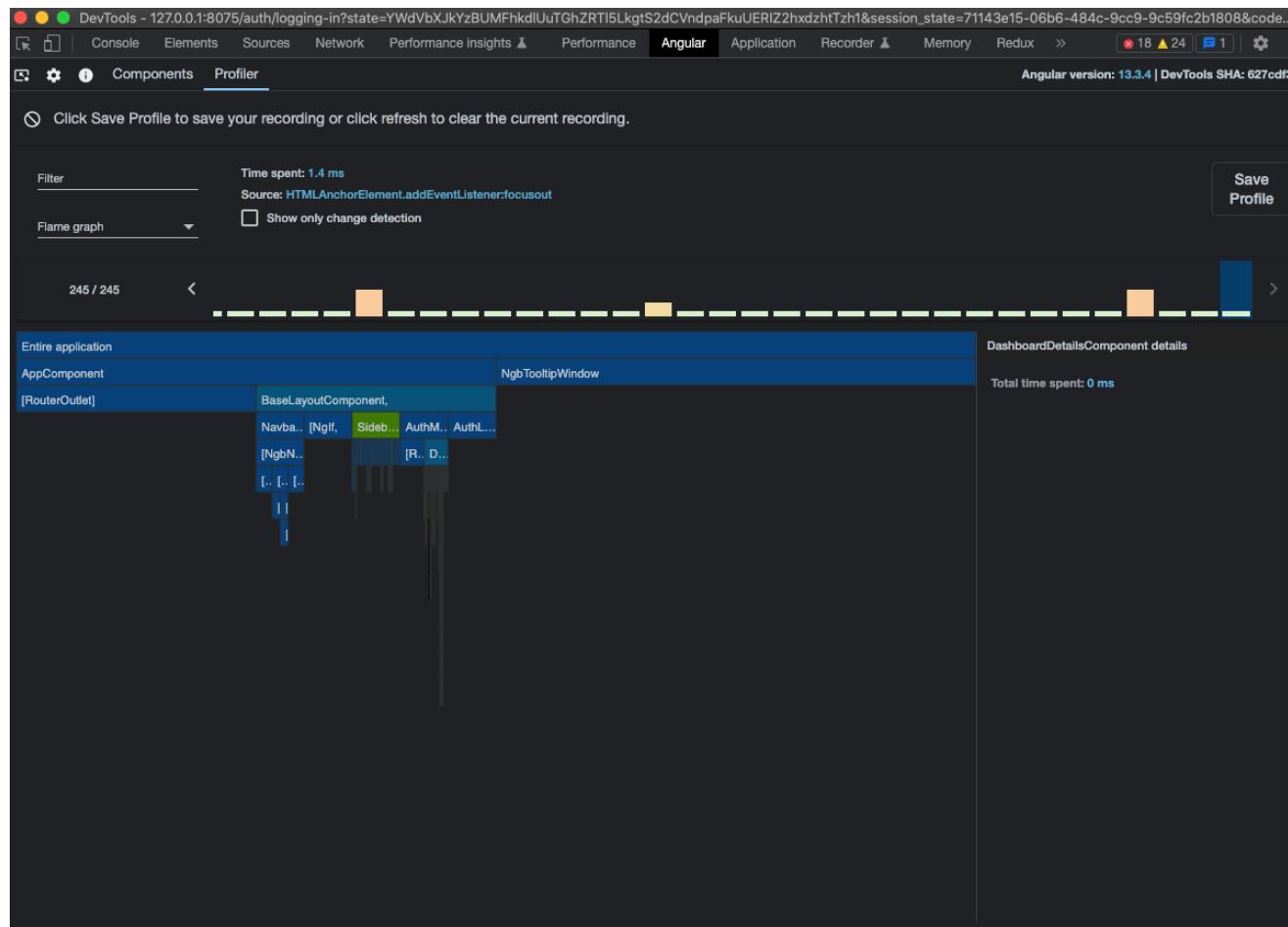
SOFTWARE
ARCHITECT

DEMO – Chrome DevTools

#4: Angular DevTools Profiler

- The Angular DevTools extension can be added to Chrome here
 - <https://chrome.google.com/webstore/detail/angular-devtools/ienfalfjdbdpebioblackkekamfmbnh>
 - It features a component tree to inspect the components and
 - A Profiler
- Profiler shows individual change detection (CD) cycles
 - What triggered CD
 - How much time it took executing CD

#4: Angular DevTools Profiler – Example



ANGULAR
ARCHITECTS

INSIDE KNOWLEDGE



SOFTWARE
ARCHITECT

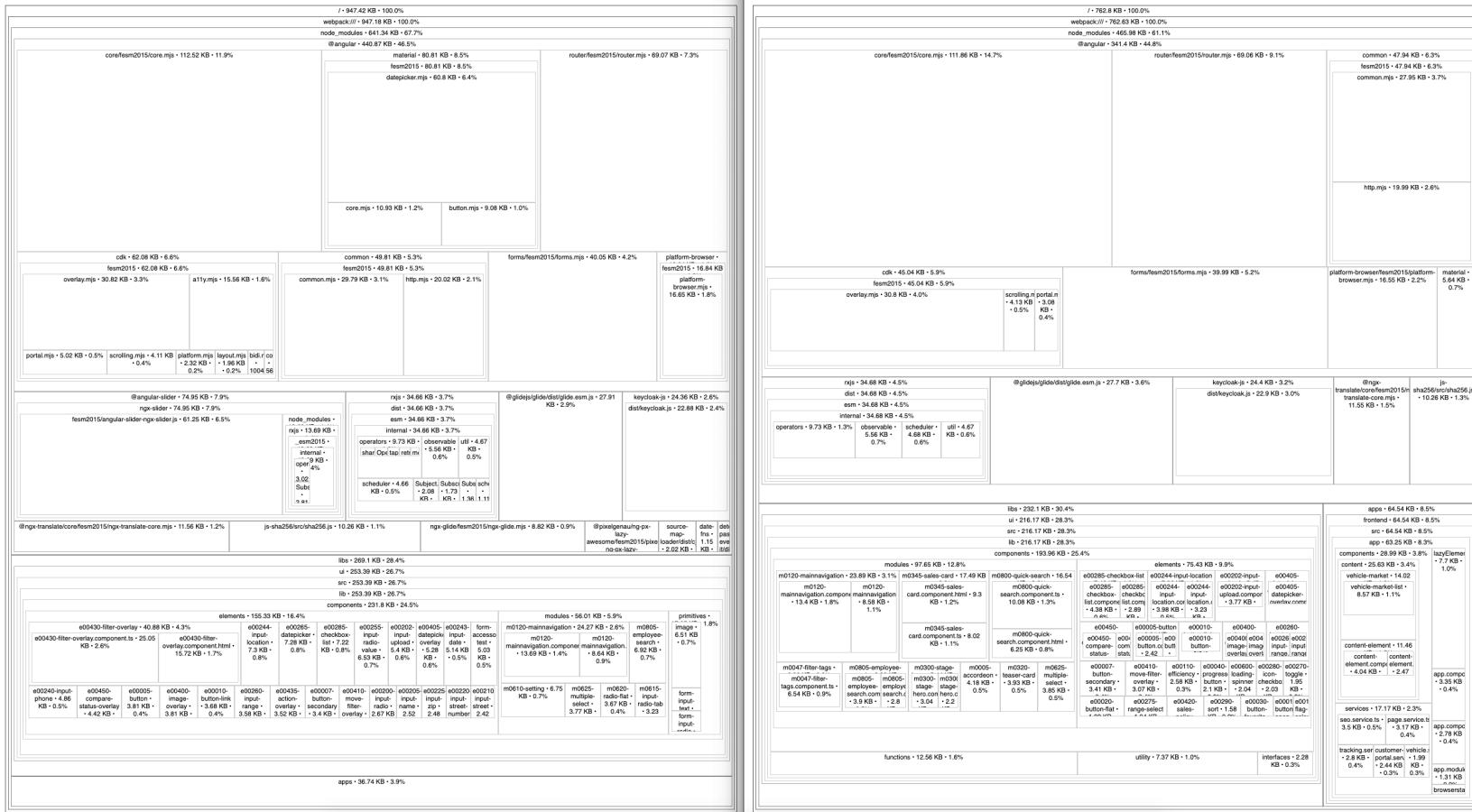
DEMO – Angular DevTools

<https://chrome.google.com/webstore/detail/angular-devtools/ienfalfjdbdpebioblfackkekamfmbnh>

#5: Source Map Explorer

- Needs a generated source map to work
 - Need to be set in build options (angular.json)
- Analyzes a single js file
 - main bundle or
 - vendor bundle (vendor chunk needs to be activated)
- Determines which file each byte in your minified code came from
- Shows you a treemap visualization where all the code is coming from

#5: Source Map Explorer (more accurate)



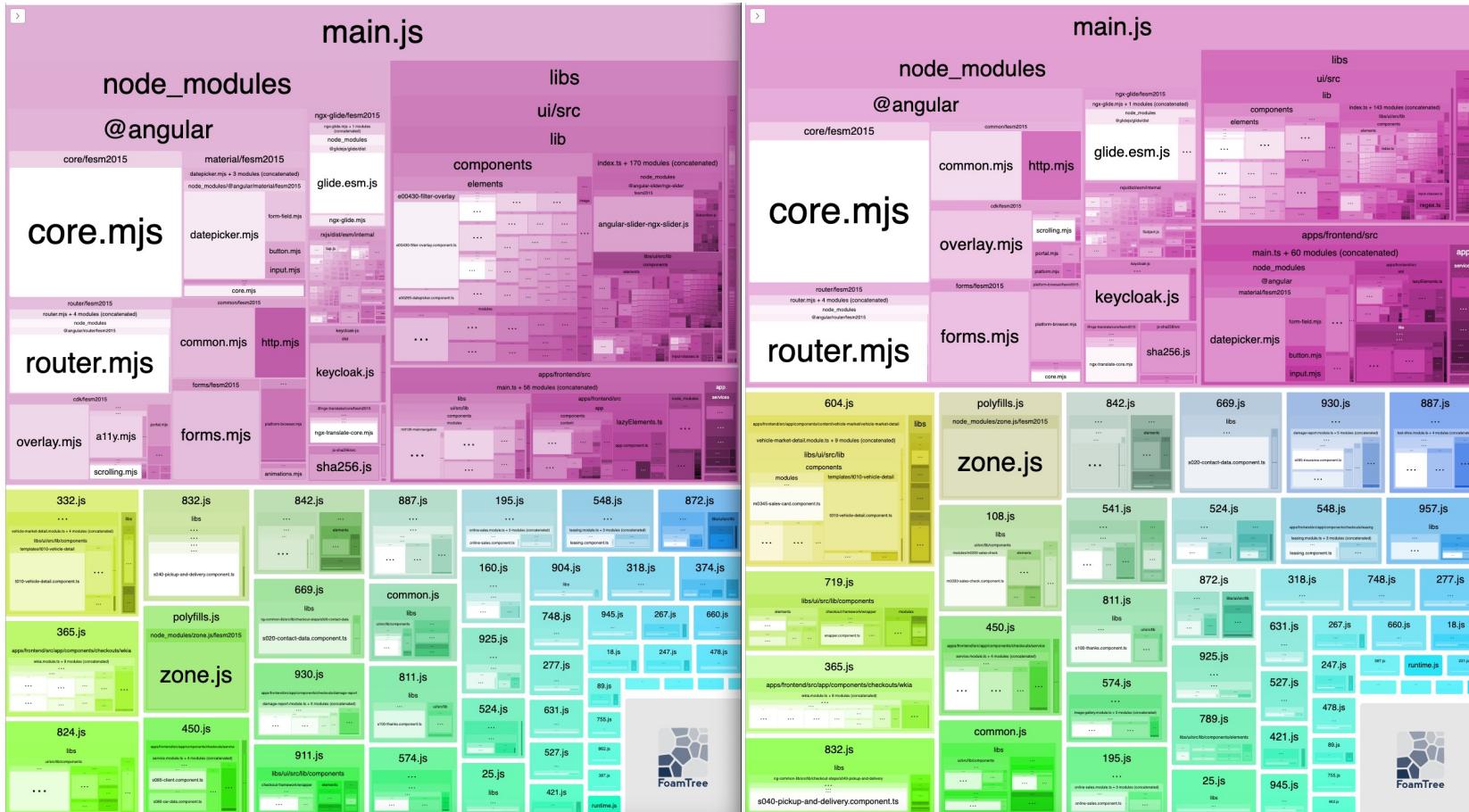
DEMO – Source Map Explorer

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

#6: Webpack Bundle Analyzer

- Needs a generated stats.json to work
 - Build flag "--stats-json"
- Analyzes the whole build
- Visualize size of all webpack output js files
 - Good to analyze lazy loading
- Interactive, zoomable and colorful treemap ☺

#6: Webpack Bundle Analyzer (colorful)



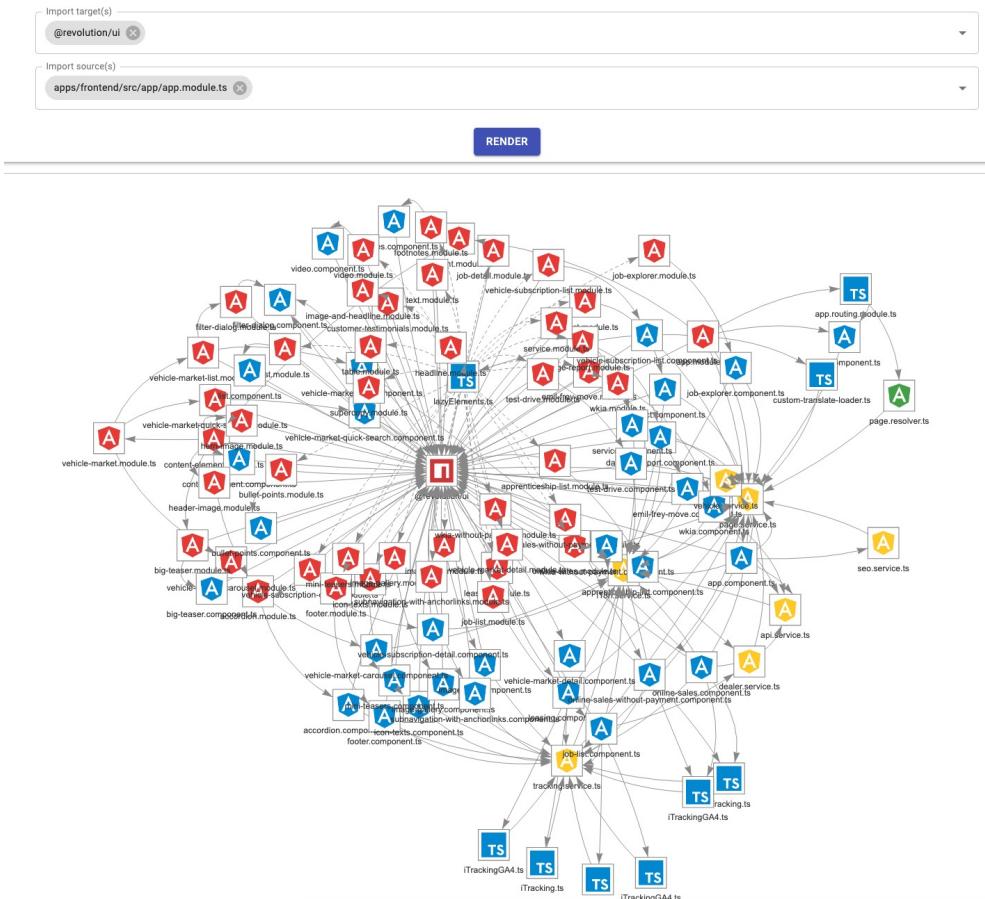
DEMO Webpack Bundle Analyzer

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

#7: Import Graph Visualizer

- The Import Graph Visualizer is a development tool for filtering and visualizing import paths within a JavaScript/TypeScript application
- Allows filtering import paths by source and target modules
- Allows you to zoom in to a limited subsection of your app, which will likely be easier to analyze than the entire app as a whole

#7: Import Graph Visualizer – Example



#7: 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#readme>

Lab

Audit Tools PageSpeed

Recap

1. Google PageSpeed Insights & Google Chrome Lighthouse
2. WebPageTest
3. Google Chrome DevTools
4. Angular DevTools Profiler
5. Sourcemap Explorer
6. Webpack Bundle Analyzer
7. Import Graph Visualizer



Initial Load

Runtime

Build Size

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>