http://localhost:3000/employees

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 50–89 90–100

METRICS

Expand view

First Contentful Paint
0.5 s

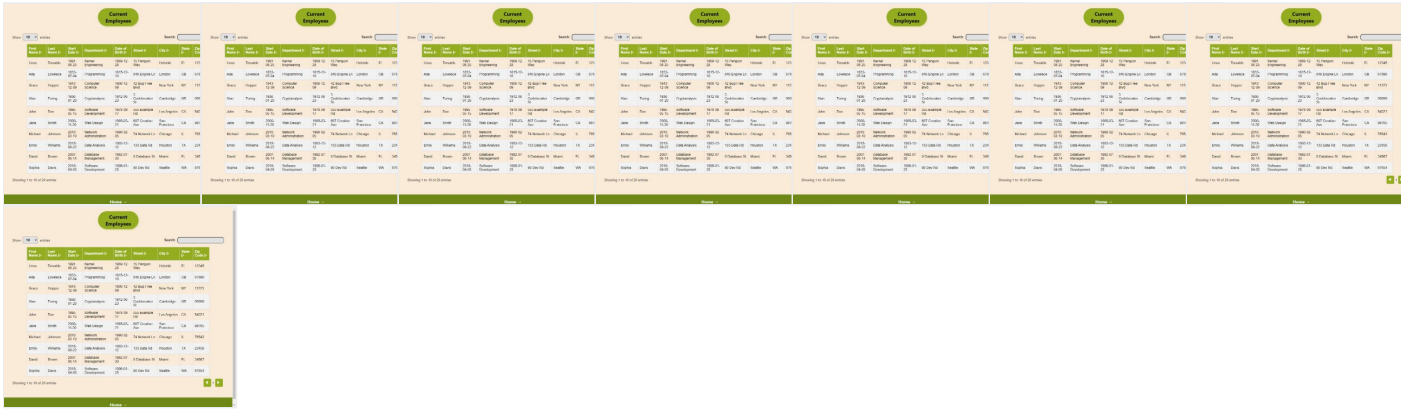
Largest Contentful Paint
0.9 s

Total Blocking Time
30 ms

Cumulative Layout Shift
0

Speed Index
0.5 s

View Treemap



Show audits relevant to: All FCP LCP TBT CLS

DIAGNOSTICS

▲ Reduce unused JavaScript — Potential savings of 341 KiB

Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity. Learn how to reduce unused JavaScript. LCP

Show 3rd-party resources (1)

URL	Transfer Size	Potential Savings
localhost 1st Party	**534.9 KiB**	**246.3 KiB**
…js/bundle.js (localhost)	534.9 KiB	246.3 KiB
F:/Openclassrooms DevApp_JS_REACT/P14/P14_OC_Wealth-Health2/wealth-health-bugfix/node_modules/react-dom/cjs/react-dom.development.js	217.1 KiB	102.2 KiB
F:/Openclassrooms DevApp_JS_REACT/P14/P14_OC_Wealth-Health2/wealth-health-bugfix/node_modules/@remix-run/router/router.ts	22.8 KiB	22.3 KiB
F:/Openclassrooms DevApp_JS_REACT/P14/P14_OC_Wealth-Health2/wealth-health-bugfix/node_modules/react/cjs/react.development.js	18.6 KiB	9.8 KiB
F:/Openclassrooms DevApp_JS_REACT/P14/P14_OC_Wealth-Health2/wealth-health-bugfix/node_modules/react-router-dom/index.tsx	8.9 KiB	7.5 KiB
F:/Openclassrooms DevApp_JS_REACT/P14/P14_OC_Wealth-Health2/wealth-health-bugfix/node_modules/react/cjs/react-jsx-dev-runtime.development.js	8.7 KiB	4.1 KiB
kaspersky-labs.com	**143.8 KiB**	**94.4 KiB**
/FD126C42-…/main.js?attr=0Hn1EkcSG… (gc.kis.v2.scr.kaspersky-labs.com)	143.8 KiB	94.4 KiB

▲ Minify JavaScript — Potential savings of 299 KiB

Minifying JavaScript files can reduce payload sizes and script parse time. Learn how to minify JavaScript. FCP LCP

Show 3rd-party resources (1)

URL	Transfer Size	Potential Savings
localhost 1st Party	**534.9 KiB**	**209.3 KiB**
…js/bundle.js (localhost)	534.9 KiB	209.3 KiB
kaspersky-labs.com	**143.8 KiB**	**52.6 KiB**
/FD126C42-…/main.js?attr=0Hn1EkcSG… (gc.kis.v2.scr.kaspersky-labs.com)	143.8 KiB	52.6 KiB
Unattributable	**89.1 KiB**	**37.1 KiB**

URL	Transfer Size	Potential Savings
chrome-extension://eimadpbcbfnmbkopoojfekhnkhdbieeh/inject/index.js	89.1 KiB	37.1 KiB

▲ Enable text compression — Potential savings of 117 KiB

Text-based resources should be served with compression (gzip, deflate or brotli) to minimize total network bytes. Learn more about text compression. FCP LCP

URL	Transfer Size	Potential Savings
kaspersky-labs.com	**143.6 KiB**	**117.4 KiB**
/FD126C42-…/main.js?attr=0Hn1EkcSG… (gc.kis.v2.scr.kaspersky-labs.com)	143.6 KiB	117.4 KiB

Serve static assets with an efficient cache policy — 1 resource found

A long cache lifetime can speed up repeat visits to your page. Learn more about efficient cache policies.

URL	Cache TTL	Transfer Size
localhost 1st Party		**535 KiB**
…js/bundle.js (localhost)	None	535 KiB

Avoid serving legacy JavaScript to modern browsers — Potential savings of 0 KiB

Polyfills and transforms enable legacy browsers to use new JavaScript features. However, many aren't necessary for modern browsers. For your bundled JavaScript, adopt a modern script deployment strategy using module/nomodule feature detection to reduce the amount of code shipped to modern browsers, while retaining support for legacy browsers. Learn how to use modern JavaScript TBT

URL		Potential Savings
localhost 1st Party		**0.1 KiB**
…js/bundle.js (localhost)		0.1 KiB
bundle.js:40010	@babel/plugin-transform-classes	
bundle.js:45400	@babel/plugin-transform-regenerator	

Page prevented back/forward cache restoration — 2 failure reasons

Many navigations are performed by going back to a previous page, or forwards again. The back/forward cache (bfcache) can speed up these return navigations. Learn more about the bfcache

Failure reason	Failure type
The page has an unload handler in the main frame. /employees (localhost)	Actionable
Pages with WebSocket cannot enter back/forward cache. /employees (localhost)	Pending browser support

Minimizes main-thread work — 0.2 s

Consider reducing the time spent parsing, compiling and executing JS. You may find delivering smaller JS payloads helps with this. Learn how to minimize main-thread work TBT

Category	Time Spent
Script Evaluation	123 ms
Other	50 ms
Script Parsing & Compilation	37 ms
Style & Layout	4 ms
Garbage Collection	4 ms
Rendering	1 ms
Parse HTML & CSS	1 ms

Avoid long main-thread tasks — 1 long task found

Lists the longest tasks on the main thread, useful for identifying worst contributors to input delay. Learn how to avoid long main-thread tasks TBT

URL	Start Time	Duration
localhost 1st Party		**75 ms**

URL	Start Time	Duration
…js/bundle.js (localhost)	958 ms	75 ms

JavaScript execution time — 0.1 s

Consider reducing the time spent parsing, compiling, and executing JS. You may find delivering smaller JS payloads helps with this. Learn how to reduce Javascript execution time. TBT

URL	Total CPU Time	Script Evaluation	Script Parse
localhost 1st Party	**117 ms**	**89 ms**	**26 ms**
…js/bundle.js (localhost)	117 ms	89 ms	26 ms

Minimize third-party usage — Third-party code blocked the main thread for 0 ms

Third-party code can significantly impact load performance. Limit the number of redundant third-party providers and try to load third-party code after your page has primarily finished loading. Learn how to minimize third-party impact. TBT

Third-Party	Transfer Size	Main-Thread Blocking Time
kaspersky-labs.com	**146 KiB**	**0 ms**
/FD126C42-…/main.js?attr=0Hn1EkcSG… (gc.kis.v2.scr.kaspersky-labs.com)	144 KiB	0 ms
Wappalyzer - Technology profiler Chrome Extension	**6 KiB**	**0 ms**

Initial server response time was short — Root document took 10 ms

Keep the server response time for the main document short because all other requests depend on it. Learn more about the Time to First Byte metric. FCP LCP

URL	Time Spent
localhost 1st Party	**10 ms**
/employees (localhost)	10 ms

Avoids enormous network payloads — Total size was 693 KiB

Large network payloads cost users real money and are highly correlated with long load times. Learn how to reduce payload sizes. LCP

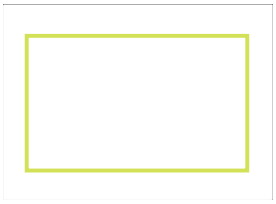
Show 3rd-party resources (5)

URL	Transfer Size
localhost 1st Party	**547.5 KiB**
…js/bundle.js (localhost)	534.9 KiB
/logo192.png (localhost)	5.6 KiB
/favicon.ico (localhost)	3.8 KiB
/employees (localhost)	2.2 KiB
/manifest.json (localhost)	0.9 KiB
kaspersky-labs.com	**145.5 KiB**
/FD126C42-…/main.js?attr=0Hn1EkcSG… (gc.kis.v2.scr.kaspersky-labs.com)	143.8 KiB
/7D8B79A2-…/init?data=eyJ1cmwiO… (gc.kis.v2.scr.kaspersky-labs.com)	1.0 KiB
…to/wsm.onHashChange?tm=2024-01-04T16%3A27%3A43.905Z (gc.kis.v2.scr.kaspersky-labs.com)	0.3 KiB
…7C0D38FD-…/from?get&nocache=… (gc.kis.v2.scr.kaspersky-labs.com)	0.3 KiB
…7C0D38FD-…/from?get&nocache=… (gc.kis.v2.scr.kaspersky-labs.com)	0.3 KiB

Avoids an excessive DOM size — 163 elements

A large DOM will increase memory usage, cause longer style calculations, and produce costly layout reflows. Learn how to avoid an excessive DOM size. TBT

Statistic	Element	Value
Total DOM Elements		163
Maximum DOM Depth	option	11
Maximum Child Elements	tbody	10



Statistic	Element	Value

Avoid chaining critical requests — 1 chain found

The Critical Request Chains below show you what resources are loaded with a high priority. Consider reducing the length of chains, reducing the download size of resources, or deferring the download of unnecessary resources to improve page load. Learn how to avoid chaining critical requests. FCP LCP

Maximum critical path latency: **25.168 ms**

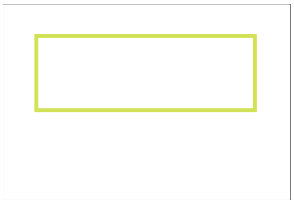
Initial Navigation

/employees (localhost)

/FD126C42-…/main.js?attr=0Hn1EkcSG… (gc.kis.v2.scr.kaspersky-labs.com) **- 2.848 ms, 143.79 KiB**

Largest Contentful Paint element — 940 ms

This is the largest contentful element painted within the viewport. Learn more about the Largest Contentful Paint element LCP

Element
 h2

Phase	% of LCP	Timing
TTFB	14%	130 ms
Load Delay	0%	0 ms
Load Time	0%	0 ms
Render Delay	86%	810 ms

More information about the performance of your application. These numbers don't directly affect the Performance score.

PASSED AUDITS (24) Hide

Eliminate render-blocking resources

Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical

JS/styles. Learn how to eliminate render-blocking resources. FCP LCP

Properly size images

Serve images that are appropriately-sized to save cellular data and improve load time. Learn how to size images.

Defer offscreen images

Consider lazy-loading offscreen and hidden images after all critical resources have finished loading to lower time to interactive. Learn how to defer offscreen images.

Minify CSS

Minifying CSS files can reduce network payload sizes. Learn how to minify CSS. FCP LCP

Reduce unused CSS

Reduce unused rules from stylesheets and defer CSS not used for above-the-fold content to decrease bytes consumed by network activity. Learn how to reduce unused CSS. FCP LCP

Efficiently encode images

Optimized images load faster and consume less cellular data. Learn how to efficiently encode images.

Serve images in next-gen formats

Image formats like WebP and AVIF often provide better compression than PNG or JPEG, which means faster downloads and less data consumption. Learn more about modern image formats.

Preconnect to required origins

Consider adding `preconnect` or `dns-prefetch` resource hints to establish early connections to important third-party origins. Learn how to preconnect to required origins. FCP LCP

Avoid multiple page redirects

Redirects introduce additional delays before the page can be loaded. Learn how to avoid page redirects. FCP LCP

Preload key requests

Consider using `<link rel=preload>` to prioritize fetching resources that are currently requested later in page load. Learn how to preload key requests. FCP LCP

Use HTTP/2

HTTP/2 offers many benefits over HTTP/1.1, including binary headers and multiplexing. Learn more about HTTP/2.

Use video formats for animated content

Large GIFs are inefficient for delivering animated content. Consider using MPEG4/WebM videos for animations and PNG/WebP for static images instead of GIF to save network bytes. Learn more about efficient video formats LCP

Remove duplicate modules in JavaScript bundles

Remove large, duplicate JavaScript modules from bundles to reduce unnecessary bytes consumed by network activity. TBT

Preload Largest Contentful Paint image

If the LCP element is dynamically added to the page, you should preload the image in order to improve LCP. Learn more about preloading LCP elements. LCP

User Timing marks and measures

Consider instrumenting your app with the User Timing API to measure your app's real-world performance during key user experiences. Learn more about User Timing marks.

All text remains visible during webfont loads

Leverage the `font-display` CSS feature to ensure text is user-visible while webfonts are loading. Learn more about `font-display`. FCP LCP

Lazy load third-party resources with facades

Some third-party embeds can be lazy loaded. Consider replacing them with a facade until they are required. Learn how to defer third-parties with a facade. TBT

Largest Contentful Paint image was not lazily loaded

Above-the-fold images that are lazily loaded render later in the page lifecycle, which can delay the largest contentful paint. Learn more about optimal lazy loading. LCP

Avoid large layout shifts

These DOM elements contribute most to the CLS of the page. Learn how to improve CLS CLS

Uses passive listeners to improve scrolling performance

Consider marking your touch and wheel event listeners as `passive` to improve your page's scroll performance. Learn more about adopting passive event listeners.

Avoids `document.write()`

For users on slow connections, external scripts dynamically injected via `document.write()` can delay page load by tens of seconds. Learn how to avoid document.write().

Avoid non-composited animations

Animations which are not composited can be janky and increase CLS. Learn how to avoid non-composited animations CLS

Image elements have explicit `width` and `height`

Set an explicit width and height on image elements to reduce layout shifts and improve CLS. Learn how to set image dimensions CLS

Has a `<meta name="viewport">` tag with `width` or `initial-scale`

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents a 300 millisecond delay to user input. Learn more about using the viewport meta tag. TBT

Captured at Jan 4, 2024, 5:27 PM GMT+1	Emulated Desktop with Lighthouse 11.2.0	Single page load
Initial page load	Custom throttling	Using Chromium 120.0.0.0 with devtools

Generated by **Lighthouse** 11.2.0 | File an issue