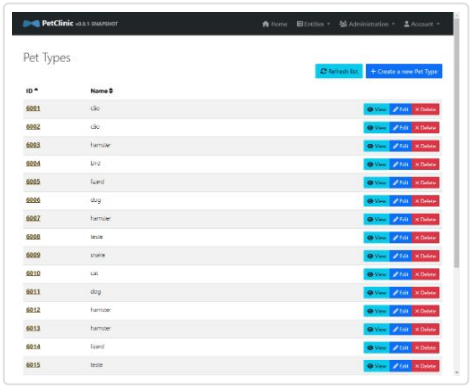
http://localhost:8080/pet-type

Performance

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

▲ 0–49 ■ 50–89 ● 90–100

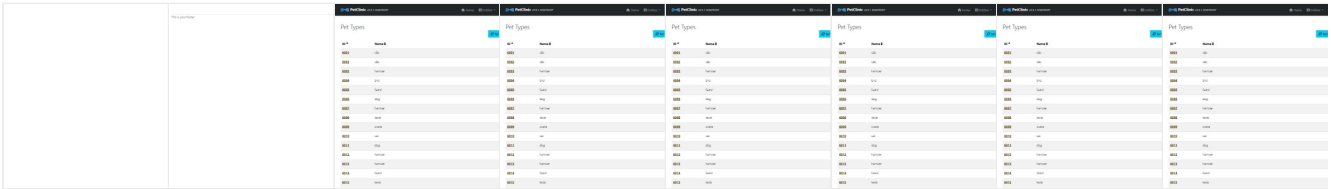


METRICS

Expand view

● First Contentful Paint 0.7 s	▲ Largest Contentful Paint 5.3 s
■ Total Blocking Time 190 ms	● Cumulative Layout Shift 0.087
● Speed Index 0.9 s	

View Treemap



Show audits relevant to: All FCP LCP TBT CLS

DIAGNOSTICS

▲ Enable text compression — Potential savings of 4,541 KiB

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

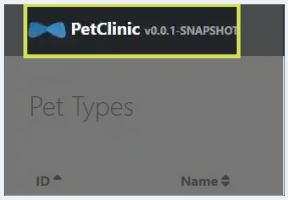
URL	Transfer size	Potential savings
localhost First Party	**5,818.9 KiB**	**4,540.8 KiB**
/src_main_webapp_bootstrap_ts.js (localhost)	5,128.9 KiB	3,968.2 KiB
/styles.css (localhost)	273.4 KiB	238.9 KiB
/src_main_webapp_app_layouts_navbar_navbar_component_ts.js (localhost)	157.3 KiB	130.0 KiB
/polyfills.js (localhost)	114.2 KiB	87.4 KiB
/common.js (localhost)	42.5 KiB	36.2 KiB
/src_main_webapp_app_entities_pet-type_list_pet-type_compone….js (localhost)	38.0 KiB	32.9 KiB
/vendor.js (localhost)	32.0 KiB	22.7 KiB
/runtime.js (localhost)	12.0 KiB	9.0 KiB
/src_main_webapp_app_entities_pet-type_pet-type_routes_ts.js (localhost)	9.1 KiB	7.2 KiB
/pet-type (localhost)	5.0 KiB	3.2 KiB
…css/loading.css (localhost)	3.4 KiB	2.7 KiB
/src_main_webapp_app_entities_entity_routes_ts.js (localhost)	3.0 KiB	2.3 KiB

▲ Largest contentful paint element — 5,290 ms

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

LCP

Element



a.navbar-brand.logo

Phase	% of LCP	Timing
TTFB	2%	130 ms
Load delay	0%	0 ms

Phase	% of LCP	Timing
Load time	0%	0 ms
Render delay	98%	5,160 ms

▲ Minify JavaScript — Potential savings of 2,032 KiB

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

URL	Transfer size	Potential savings
localhost First Party	**5,367.7 KiB**	**2,032.1 KiB**
/src_main_webapp_bootstrap_ts.js (localhost)	5,128.9 KiB	1,943.6 KiB
/polyfills.js (localhost)	114.2 KiB	49.5 KiB
/vendor.js (localhost)	32.0 KiB	19.0 KiB
/common.js (localhost)	42.5 KiB	9.0 KiB
/runtime.js (localhost)	12.0 KiB	6.4 KiB
/src_main_webapp_app_entities_pet-type_list_pet-type_compone….js (localhost)	38.0 KiB	4.6 KiB

▲ Reduce unused JavaScript — Potential savings of 1,684 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. FCP LCP

URL	Transfer size	Potential savings
localhost First Party	**5,162.4 KiB**	**1,683.9 KiB**
/src_main_webapp_bootstrap_ts.js (localhost)	5,120.2 KiB	1,658.3 KiB
…node_modules/@angular/core/fesm2022/core.mjs	1,380.3 KiB	480.8 KiB
…node_modules/@ng-bootstrap/ng-bootstrap/fesm2022/ng-bootstrap.mjs	587.2 KiB	331.8 KiB
…node_modules/@angular/common/fesm2022/common.mjs	284.9 KiB	127.5 KiB
…node_modules/@angular/forms/fesm2022/forms.mjs	272.0 KiB	116.3 KiB
…node_modules/@angular/router/fesm2022/router.mjs	278.7 KiB	58.2 KiB
/common.js (localhost)	42.2 KiB	25.6 KiB

URL	Transfer size	Potential savings
…entities/pet-type/update/pet-type-update.component.html	7.4 KiB	7.4 KiB
…entities/pet-type/update/pet-type-update.component.ts	3.4 KiB	3.4 KiB
…layouts/error/error.component.html	2.2 KiB	2.2 KiB
…entities/pet-type/update/pet-type-form.service.ts	1.0 KiB	1.0 KiB
…layouts/error/error.component.ts	0.8 KiB	0.8 KiB

▲ Eliminate render-blocking resources — Potential savings of 220 ms

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

URL	Transfer size	Potential savings
localhost First Party	**274.2 KiB**	**530 ms**
/styles.css (localhost)	274.2 KiB	530 ms

▲ Minify CSS — Potential savings of 69 KiB

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

URL	Transfer size	Potential savings
localhost First Party	**274.2 KiB**	**48.2 KiB**
/styles.css (localhost)	274.2 KiB	48.2 KiB
Unattributable	**20.9 KiB**	**20.8 KiB**
`.navbar-version[_ngcontent-ng-c50362896] { font-size: 0.65em; color: rgba(255, 255, 255, 0.55); … } …`	20.9 KiB	20.8 KiB

▲ Reduce unused CSS — Potential savings of 271 KiB

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

URL	Transfer size	Potential savings
localhost First Party	**273.4 KiB**	**250.7 KiB**
/styles.css (localhost)	273.4 KiB	250.7 KiB
Unattributable	**20.9 KiB**	**20.8 KiB**
`.navbar-version[_ngcontent-ng-c50362896] { font-size: 0.65em; color: rgba(255, 255, 255, 0.55); … } …`	20.9 KiB	20.8 KiB

▲ 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

Reason for failure	Failure type
Pages whose main resource has cache-control:no-store cannot enter back-forward cache. /pet-type (localhost)	Not actionable
JsNetworkRequestReceivedCacheControlNoStoreResource /pet-type (localhost)	Not actionable

■ Avoid enormous network payloads — Total size was 5,847 KiB

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

URL	Transfer size
localhost First Party	**5,822.4 KiB**
/src_main_webapp_bootstrap_ts.js (localhost)	5,129.7 KiB
/styles.css (localhost)	274.2 KiB
/src_main_webapp_app_layouts_navbar_navbar_component_ts.js (localhost)	158.1 KiB
/polyfills.js (localhost)	115.0 KiB
/common.js (localhost)	43.3 KiB
/src_main_webapp_app_entities_pet-type_list_pet-type_compone….js (localhost)	38.8 KiB

URL	Transfer size
/vendor.js (localhost)	32.8 KiB
/runtime.js (localhost)	12.8 KiB
/src_main_webapp_app_entities_pet-type_pet-type_routes_ts.js (localhost)	9.9 KiB
…images/jhipster_family_member_1_head-192.png (localhost)	7.7 KiB

JavaScript execution time — 0.6 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 First Party	**713 ms**	**580 ms**	**1 ms**
/polyfills.js (localhost)	382 ms	358 ms	0 ms
/runtime.js (localhost)	172 ms	168 ms	0 ms
/pet-type (localhost)	159 ms	54 ms	1 ms
Unattributable	**143 ms**	**8 ms**	**0 ms**
`Unattributable`	143 ms	8 ms	0 ms

Minimises main-thread work — 0.9 s

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

Category	Time Spent
Script Evaluation	591 ms
Other	213 ms
Parse HTML & CSS	32 ms
Style & Layout	29 ms

Category	Time Spent
Script Parsing & Compilation	9 ms
Rendering	5 ms
Garbage Collection	2 ms

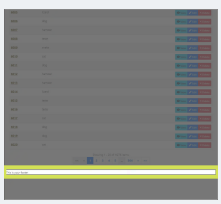
Avoid long main-thread tasks — 5 long tasks found

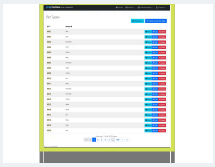
Lists the longest tasks on the main thread – useful for identifying worst contributors to input delay. Learn how to avoid long main-thread tasks TBT

URL	Start Time	Duration
localhost First Party		**504 ms**
/runtime.js (localhost)	612 ms	145 ms
/polyfills.js (localhost)	4,831 ms	129 ms
/polyfills.js (localhost)	4,960 ms	124 ms
/polyfills.js (localhost)	5,306 ms	106 ms
Unattributable		**68 ms**
`Unattributable`	204 ms	68 ms

Avoid large layout shifts — 3 layout shifts found

These are the largest layout shifts observed on the page. Each table item represents a single layout shift, and shows the element that shifted the most. Below each item are possible root causes that led to the layout shift. Some of these layout shifts may not be included in the CLS metric value due to windowing. Learn how to improve CLS CLS

Element	Layout shift score
 div.footer	0.054
/styles.css (localhost)	A late network request adjusted the page layout
…css/loading.css (localhost)	A late network request adjusted the page layout
 body	0.029

Element	Layout shift score
	
 body	0.004

User Timing marks and measures — 41 user timings

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.

Name	Type	Start Time	Duration
Zone	Measure	90.24 ms	1.26 ms
Zone:ZoneAwarePromise	Measure	92.71 ms	1.59 ms
Zone:toString	Measure	94.57 ms	0.23 ms
Zone:util	Measure	95.06 ms	1.14 ms
Zone:legacy	Measure	96.71 ms	0.19 ms
Zone:timers	Measure	97.03 ms	0.57 ms
Zone:requestAnimationFrame	Measure	97.70 ms	0.20 ms
Zone:blocking	Measure	98.07 ms	0.23 ms
Zone:EventTarget	Measure	98.42 ms	0.88 ms
Zone:MutationObserver	Measure	99.48 ms	0.22 ms
Zone:IntersectionObserver	Measure	99.88 ms	0.12 ms
Zone:FileReader	Measure	100.19 ms	0.11 ms
Zone:on_property	Measure	100.50 ms	28.50 ms
Zone:customElements	Measure	129.16 ms	0.34 ms
Zone:XHR	Measure	129.63 ms	0.37 ms
Zone:geolocation	Measure	130.20 ms	0.20 ms

Name	Type	Start Time	Duration
Zone:PromiseRejectionEvent	Measure	130.49 ms	0.11 ms
Zone:queueMicrotask	Measure	130.75 ms	0.05 ms
Zone	Mark	90.24 ms	
Zone:ZoneAwarePromise	Mark	92.71 ms	
Zone:toString	Mark	94.57 ms	
Zone:util	Mark	95.06 ms	
Zone:legacy	Mark	96.71 ms	
Zone:timers	Mark	97.03 ms	
Zone:requestAnimationFrame	Mark	97.70 ms	
Zone:blocking	Mark	98.07 ms	
Zone:EventTarget	Mark	98.42 ms	
Zone:MutationObserver	Mark	99.48 ms	
Zone:IntersectionObserver	Mark	99.88 ms	
Zone:FileReader	Mark	100.19 ms	
Zone:on_property	Mark	100.50 ms	
Zone:customElements	Mark	129.16 ms	
Zone:XHR	Mark	129.63 ms	
Zone:geolocation	Mark	130.20 ms	
Zone:PromiseRejectionEvent	Mark	130.49 ms	
Zone:queueMicrotask	Mark	130.75 ms	
mark_feature_usage	Mark	402.24 ms	
mark_feature_usage	Mark	455.38 ms	
mark_feature_usage	Mark	742.31 ms	
mark_feature_usage	Mark	811.00 ms	

Name	Type	Start Time	Duration
mark_feature_usage	Mark	819.39 ms	

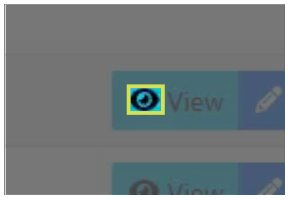
Initial server response time was short — Root document took 10 ms

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 First Party	**10 ms**
/pet-type (localhost)	10 ms

Avoids an excessive DOM size — 623 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		623
Maximum DOM Depth	 path	17
Maximum Child Elements	 tbody	20

Avoid chaining critical requests — 8 chains 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.

Maximum critical path latency: **719.85 ms**

Initial Navigation

- /pet-type (localhost)
  - …css/loading.css (localhost) **- 7.075 ms, 4.20 KiB**

/styles.css (localhost) **- 12.835 ms, 274.17 KiB**
/runtime.js (localhost) **- 22.156 ms, 12.82 KiB**
/polyfills.js (localhost)
/src_main_webapp_bootstrap_ts.js (localhost)
/src_main_webapp_app_entities_entity_routes_ts.js (localhost)
/common.js (localhost) **- 5.863 ms, 43.34 KiB**
/src_main_webapp_app_entities_pet-type_pet-type_routes_ts.js (localhost)
/src_main_webapp_app_entities_pet-type_list_pet-type_compone….js (localhost) **- 4.898 ms, 38.83 I**
/src_main_webapp_app_layouts_navbar_navbar_component_ts.js (localhost) **- 6.699 ms, 158.12 KiB**
/vendor.js (localhost) **- 17.578 ms, 32.82 KiB**
/main.js (localhost) **- 13.762 ms, 1.67 KiB**

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

PASSED AUDITS (21) Hide

Properly size images

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

Defer off-screen 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. FCP LCP

Efficiently encode images

Optimised images load faster and consume less mobile data. Learn how to efficiently encode images. FCP LCP

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. FCP LCP

Pre-connect 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. LCP FCP

Avoid multiple page redirects

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

Use HTTP/2

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

- 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 FCP LCP

- Remove duplicate modules in JavaScript bundles

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

- Avoid serving legacy JavaScript to modern browsers

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 delivered to modern browsers, while retaining support for legacy browsers. Learn how to use modern JavaScript FCP LCP

- 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

- Uses efficient cache policy on static assets — 0 resources found

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

- All text remains visible during webfont loads

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

- Minimise third-party usage

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 minimise third-party impact. TBT

- 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

- 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 that are not composited can be poor, slow 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 optimises your app for mobile screen sizes, but also prevents a 300 millisecond delay to user input. Learn more about using the viewport meta tag.

Captured at 5 Dec 2024, 16:12 WET
Initial page load
Emulated desktop with Lighthouse 12.2.1
Custom throttling
Single-page session
Using Chromium 131.0.0.0 with devtools

Generated by **Lighthouse** 12.2.1 | File an issue