

Performance Metrics First Meaningful Paint First Contentful Paint 760 ms 🐼 760 ms 🐼 760 ms 🐼 First CPU Idle Speed Index 760 ms 🗸 Time to Interactive 760 ms 🗸 Estimated Input Latency 13 ms 🐼 Values are estimated and may vary.

Opportunities

These are opportunities to speed up your application by optimizing the following resources.

Resource to optimize	Esti	mated Savings	
Eliminate render-blocking resources		0,16 s ^	
Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. <u>Learn more</u> .			
URL	Size (KB)	Download Time (ms)	
	(ICD)		

Q Diagnostics

More information about the performance of your application.

Uses efficient cache policy on static assets

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

4 assets found

URL	Cache TTL	Size (KB)
/styles/inline.css (127.0.0.1)	None	12 KB
/scripts/app.js (127.0.0.1)	None	7 KB
/images/ic_refresh_white_24px.svg (127.0.0.1)	None	0 KB
/images/ic_add_white_24px.svg (127.0.0.1)	None	0 KB

2 Critical Request Chains

1 chains found

The Critical Request Chains below show you what resources are issued 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 more.

Longest chain: 45,2ms over 2 requests, totalling 12,5 KB

Initial Navigation

/ (127.0.0.1)

/styles/inline.css (127.0.0.1) - 22,3ms, 12,47 KB

19 audits ^ Passed audits Properly size images

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

2 Defer offscreen images



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

Potential savings of 3 KB <a> ^ 3 Minify CSS

Minifying CSS files can reduce network payload sizes. Learn more.

Potential **URL** Original Savings

/styles/inline.css (127.0.0.1) 3 KB 12 KB

Minify JavaScript

Potential savings of 4 KB <

Minifying JavaScript files can reduce payload sizes and script parse time. Learn more.

Potential **URL** Original Savings /scripts/app.js (127.0.0.1) 7 KB 4 KB

5 Defer unused CSS

Potential savings of 7 KB <



Remove unused rules from stylesheets to reduce unnecessary bytes consumed by network activity. Learn more.

Potential URL Original Savings

Potential **URL** Original Savings 12 KB 7 KB /styles/inline.css (127.0.0.1) Efficiently encode images Optimized images load faster and consume less cellular data. Learn more. 7 Serve images in next-gen formats Image formats like JPEG 2000, JPEG XR, and WebP often provide better compression than PNG or JPEG, which means faster downloads and less data consumption. Learn more. Enable text compression Potential savings of 17 KB < Text-based responses should be served with compression (gzip, deflate or brotli) to minimize total network bytes. Learn more. **GZIP** Uncompressed resource URL Original Savings /styles/inline.css (127.0.0.1) 12 KB 10 KB /scripts/app.js (127.0.0.1) 6 KB 5 KB http://127.0.0.1:8887 4 KB 3 KB Avoid multiple, costly round trips to any origin Potential savings of 0 ms < Consider adding preconnect or dns-prefetch resource hints to establish early connections to important third-party origins. Learn more. 10 Keep server response times low (TTFB) Time To First Byte identifies the time at which your server sends a response. Learn more. 0 ms 11 Avoid multiple page redirects Redirects introduce additional delays before the page can be loaded. Learn more. 12 Preload key requests Potential savings of 0 ms < Consider using k rel=preload> to prioritize fetching late-discovered resources sooner. Learn more. 13 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 Total size was 25 KB 14 Avoids enormous network payloads Large network payloads cost users real money and are highly correlated with long load times. Learn more.

URL	Total Size	Transfer Time
/styles/inline.css (127.0.0.1)	12 KB	10 ms
/scripts/app.js (127.0.0.1)	7 KB	0 ms
http://127.0.0.1:8887	4 KB	0 ms
bastille/A (api-ratp.pierre-grimaud.fr)	1 KB	0 ms
/images/ic_refresh_white_24px.svg (127.0.0.1)	0 KB	0 ms
/images/ic_add_white_24px.svg (127.0.0.1)	0 KB	0 ms

15 Avoids an excessive DOM size

65 nodes ♥ ^



Browser engineers recommend pages contain fewer than ~1.500 DOM nodes. The sweet spot is a tree depth < 32 elements and fewer than 60 children/parent element. A large DOM can increase memory usage, cause longer style calculations, and produce costly layout reflows. Learn more.

Total DOM Nodes	Maximum DOM Depth	Maximum Children
65	7	12
	<pre><option value="metros/1/nation/A"></option></pre>	<pre><select id="selectTimetableToAdd"></select></pre>

16 User Timing marks and measures



Consider instrumenting your app with the User Timing API to create custom, real-world measurements of key user experiences. Learn more.

17 JavaScript boot-up time





Consider reducing the time spent parsing, compiling, and executing JS. You may find delivering smaller JS payloads helps with this. Learn more.

18 Minimizes main thread work

350 ms 🐶 🔨



Consider reducing the time spent parsing, compiling and executing JS. You may find delivering smaller JS payloads helps with this.

Time Spent	Category
178 ms	Other
94 ms	Script Evaluation
34 ms	Garbage Collection
13 ms	Style & Layout
13 ms	Rendering
7 ms	Script Parsing & Compilation
6 ms	Parse HTML & CSS

19 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.

Progressive Web App

These checks validate the aspects of a Progressive Web App, as specified by the baseline PWA Checklist.



1 Does not respond with a 200 when offline



If you're building a Progressive Web App, consider using a service worker so that your app can work offline. <u>Learn</u> more.

2 User will not be prompted to Install the Web App



Failures: No manifest was fetched, Site does not register a service worker.

Browsers can proactively prompt users to add your app to their homescreen, which can lead to higher engagement. <u>Learn more</u>.

3 Does not redirect HTTP traffic to HTTPS



If you've already set up HTTPS, make sure that you redirect all HTTP traffic to HTTPS. Learn more.

4 Does not register a service worker



The service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. Learn more.

5 Is not configured for a custom splash screen



Failures: No manifest was fetched.

A themed splash screen ensures a high-quality experience when users launch your app from their homescreens. <u>Learn more</u>.

6 Address bar does not match brand colors



Failures: No manifest was fetched, No `<meta name="theme-color">` tag found.

The browser address bar can be themed to match your site. <u>Learn more</u>.

Additional items to manually check



These checks are required by the baseline <u>PWA Checklist</u> but are not automatically checked by Lighthouse. They do not affect your score but it's important that you verify them manually.

1 Site works cross-browser



To reach the most number of users, sites should work across every major browser. Learn more.

2 Page transitions don't feel like they block on the network



Transitions should feel snappy as you tap around, even on a slow network, a key to perceived performance. <u>Learn</u> more.

3 Each page has a URL



Ensure individual pages are deep linkable via the URLs and that URLs are unique for the purpose of shareability on social media. <u>Learn more</u>.

✓ Passed audits



1 Page load is fast enough on 3G



A fast page load over a 3G network ensures a good mobile user experience. Learn more.

2 Uses HTTPS

All sites should be protected with HTTPS, even ones that don't handle sensitive data. HTTPS prevents intruders from tampering with or passively listening in on the communications between your app and your users, and is a prerequisite for HTTP/2 and many new web platform APIs. Learn more.

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

Add a viewport meta tag to optimize your app for mobile screens. Learn more.

4 Contains some content when JavaScript is not available

Your app should display some content when JavaScript is disabled, even if it's just a warning to the user that JavaScript is required to use the app. <u>Learn more</u>.

5 Content is sized correctly for the viewport

If the width of your app's content doesn't match the width of the viewport, your app might not be optimized for mobile screens. <u>Learn more</u>.

Not applicable

1 audits ^

The short_name won't be truncated on the homescreen



Make your app's `short_name` fewer than 12 characters to ensure that it's not truncated on homescreens. <u>Learn more</u>.

Accessibility



These checks highlight opportunities to <u>improve the accessibility of your web app</u>. Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

Elements Describe Contents Well

These are opportunities to make your content easier to understand for a user of assistive technology, like a screen reader.

1 Form elements do not have associated labels



Labels ensure that form controls are announced properly by assistive technologies, like screen readers. <u>Learn</u> more.

Failing Elements

<select id="selectTimetableToAdd">

Color Contrast Is Satisfactory

These are opportunities to improve the legibility of your content.

Background and foreground colors do not have a sufficient contrast ratio.



Low-contrast text is difficult or impossible for many users to read. Learn more.

Failing Elements

<h1 class="header title">Paris Métro 1</h1>

Page Specifies Valid Language

These are opportunities to improve the interpretation of your content by users in different locales.

(html> element does not have a [lang] attribute



If a page doesn't specify a lang attribute, a screen reader assumes that the page is in the default language that the user chose when setting up the screen reader. If the page isn't actually in the default language, then the screen reader might not announce the page's text correctly. <u>Learn more</u>.

Failing Elements

<html>

Additional items to manually check 10 audits ^ These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review. 1 The page has a logical tab order Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. Learn more. 2 Interactive controls are keyboard focusable Custom interactive controls are keyboard focusable and display a focus indicator. Learn more. The user's focus is directed to new content added to the page If new content, such as a dialog, is added to the page, the user's focus is directed to it. Learn more. User focus is not accidentally trapped in a region A user can tab into and out of any control or region without accidentally trapping their focus. Learn more. 5 Custom controls have associated labels Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. Learn more. 6 Custom controls have ARIA roles Custom interactive controls have appropriate ARIA roles. Learn more. 7 Visual order on the page follows DOM order DOM order matches the visual order, improving navigation for assistive technology. Learn more. 8 Offscreen content is hidden from assistive technology Offscreen content is hidden with display: none or aria-hidden=true. Learn more. Headings don't skip levels Headings are used to create an outline for the page and heading levels are not skipped. Learn more. 10 HTML5 landmark elements are used to improve navigation Landmark elements (<main>, <nav>, etc.) are used to improve the keyboard navigation of the page for assistive technology. Learn more. 7 audits ^ ✓ Passed audits **ARIA Attributes Follow Best Practices** These are opportunities to improve the usage of ARIA in your application which may enhance the experience for users of assistive technology, like a screen reader. [aria-*] attributes match their roles Each ARIA 'role' supports a specific subset of 'aria-*' attributes. Mismatching these invalidates the 'aria-*' attributes. Learn more. 2 [aria-*] attributes have valid values

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid values. Learn more.

[aria-*] attributes are valid and not misspelled



Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid names. Learn more.

Elements Have Discernible Names



These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

Buttons have an accessible name



When a button doesn't have an accessible name, screen readers announce it as "button", making it unusable for users who rely on screen readers. Learn more.

Elements Describe Contents Well



These are opportunities to make your content easier to understand for a user of assistive technology, like a screen reader.

Document has a <title> element



The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. Learn more.

Elements Are Well Structured



These are opportunities to make sure your HTML is appropriately structured.

[id] attributes on the page are unique 1

scale] attribute is not less than 5.



The value of an id attribute must be unique to prevent other instances from being overlooked by assistive technologies. Learn more.

Meta Tags Used Properly



These are opportunities to improve the user experience of your site.

[user-scalable="no"] is not used in the <meta name="viewport"> element and the [maximum-



Disabling zooming is problematic for users with low vision who rely on screen magnification to properly see the contents of a web page. Learn more.

Not applicable

25 audits ^

Elements Use Attributes Correctly

These are opportunities to improve the configuration of your HTML elements.

1 [accesskey] values are unique



Access keys let users quickly focus a part of the page. For proper navigation, each access key must be unique. Learn more.

<audio> elements contain a <track> element with [kind="captions"]



Captions make audio elements usable for deaf or hearing-impaired users, providing critical information such as who is talking, what they're saying, and other non-speech information. Learn more.

Image elements have [alt] attributes



Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. <u>Learn more</u>.

4 <input type="image"> elements have [alt] text

When an image is being used as an `<input>` button, providing alternative text can help screen reader users understand the purpose of the button. <u>Learn more</u>.

5 No element has a [tabindex] value greater than 0



A value greater than 0 implies an explicit navigation ordering. Although technically valid, this often creates frustrating experiences for users who rely on assistive technologies. <u>Learn more</u>.

6 Cells in a element that use the [headers] attribute only refer to other cells of that same table.



Screen readers have features to make navigating tables easier. Ensuring `` cells using the `[headers]` attribute only refer to other cells in the same table may improve the experience for screen reader users. <u>Learn more</u>.

7 elements and elements with [role="columnheader"/"rowheader"] have data cells they describe.



Screen readers have features to make navigating tables easier. Ensuring table headers always refer to some set of cells may improve the experience for screen reader users. Learn more.

ARIA Attributes Follow Best Practices



These are opportunities to improve the usage of ARIA in your application which may enhance the experience for users of assistive technology, like a screen reader.

1 [role]s have all required [aria-*] attributes



Some ARIA roles have required attributes that describe the state of the element to screen readers. Learn more.

Elements with [role] that require specific children [role]s, are present



Some ARIA parent roles must contain specific child roles to perform their intended accessibility functions. <u>Learn more</u>.

3 [role]s are contained by their required parent element



Some ARIA child roles must be contained by specific parent roles to properly perform their intended accessibility functions. <u>Learn more</u>.

4 [role] values are valid



ARIA roles must have valid values in order to perform their intended accessibility functions. Learn more.

Elements Have Discernible Names



These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

1 Links have a discernible name



Link text (and alternate text for images, when used as links) that is discernible, unique, and focusable improves the navigation experience for screen reader users. <u>Learn more</u>.

Elements Describe Contents Well



These are opportunities to make your content easier to understand for a user of assistive technology, like a screen reader.

1 The page contains a heading, skip link, or landmark region



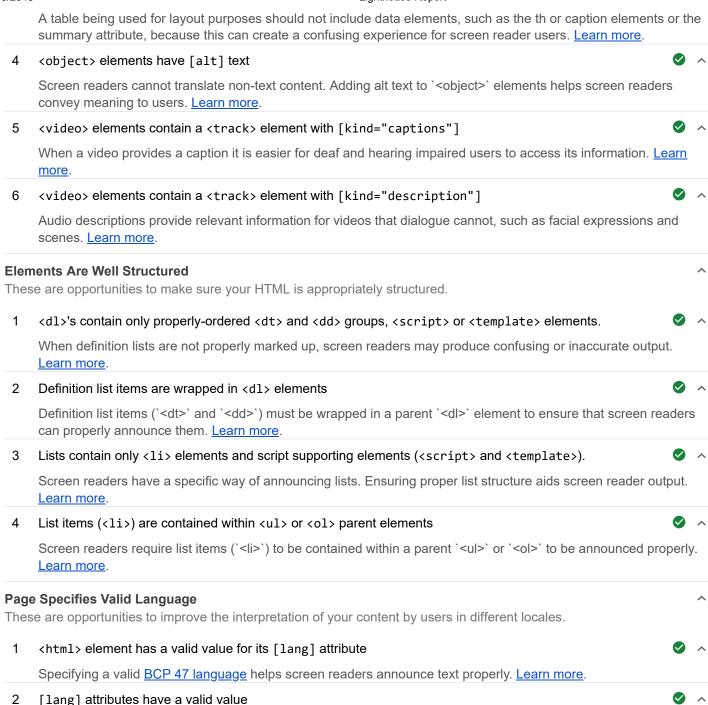
Adding ways to bypass repetitive content lets keyboard users navigate the page more efficiently. Learn more.

<frame> or <iframe> elements have a title



Screen reader users rely on frame titles to describe the contents of frames. Learn more.

Presentational elements avoid using , <caption> or the [summary] attribute.



Meta Tags Used Properly

reader. Learn more.

These are opportunities to improve the user experience of your site.

1 The document does not use <meta http-equiv="refresh">

Users do not expect a page to refresh automatically, and doing so will move focus back to the top of the page. This may create a frustrating or confusing experience. <u>Learn more</u>.

Specifying a valid BCP 47 language on elements helps ensure that text is pronounced correctly by a screen

Best Practices



Does not use HTTP/2 for all of its resources 1

5 requests not served via HTTP/2 A

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

URL	Protocol
http://127.0.0.1:8887	http/1.1
/styles/inline.css (127.0.0.1)	http/1.1
/scripts/app.js (127.0.0.1)	http/1.1
/images/ic_add_white_24px.svg (127.0.0.1)	http/1.1
/images/ic_refresh_white_24px.svg (127.0.0.1)	http/1.1

Passed audits

14 audits

Avoids Application Cache

Application Cache is deprecated. Learn more.

Avoids WebSQL DB

Web SQL is deprecated. Consider using IndexedDB instead. Learn more.

3 **Uses HTTPS**

All sites should be protected with HTTPS, even ones that don't handle sensitive data. HTTPS prevents intruders from tampering with or passively listening in on the communications between your app and your users, and is a prerequisite for HTTP/2 and many new web platform APIs. Learn more.

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.

Avoids document.write()



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

Links to cross-origin destinations are safe



Add `rel="noopener"` or `rel="noreferrer"` to any external links to improve performance and prevent security vulnerabilities. Learn more.

Avoids requesting the geolocation permission on page load



Users are mistrustful of or confused by sites that request their location without context. Consider tying the request to user gestures instead. Learn more.

Page has the HTML doctype



Specifying a doctype prevents the browser from switching to quirks-mode. Read more on the MDN Web Docs page

Avoids front-end JavaScript libraries with known security vulnerabilities



Some third-party scripts may contain known security vulnerabilities that are easily identified and exploited by attackers. Learn more.

10 Avoids requesting the notification permission on page load



16/8/2018 Lighthouse Report

> Users are mistrustful of or confused by sites that request to send notifications without context. Consider tying the request to user gestures instead. Learn more.

11 Avoids deprecated APIs



Deprecated APIs will eventually be removed from the browser. Learn more.

12 Allows users to paste into password fields



Preventing password pasting undermines good security policy. Learn more.

13 No browser errors logged to the console



Errors logged to the console indicate unresolved problems. They can come from network request failures and other browser concerns.

14 Displays images with correct aspect ratio



Image display dimensions should match natural aspect ratio. Learn more.

SEO



These checks ensure that your page is optimized for search engine results ranking. There are additional factors Lighthouse does not check that may affect your search ranking. Learn more.

Content Best Practices

Format your HTML in a way that enables crawlers to better understand your app's content.

Document does not have a meta description



Meta descriptions may be included in search results to concisely summarize page content. Learn more.

2 Document does not have a valid rel=canonical



Points to a different domain (https://weather-pwa-sample.firebaseapp.com/final/)

Canonical links suggest which URL to show in search results. Learn more.

Additional items to manually check



Run these additional validators on your site to check additional SEO best practices.

Page is mobile friendly



Take the Mobile-Friendly Test to check for audits not covered by Lighthouse, like sizing tap targets appropriately. Learn more.

Structured data is valid



Run the Structured Data Testing Tool and the Structured Data Linter to validate structured data. Learn more.



Passed audits

8 audits ^

Mobile Friendly



Make sure your pages are mobile friendly so users don't have to pinch or zoom in order to read the content pages. Learn more.

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



Add a viewport meta tag to optimize your app for mobile screens. Learn more.

16/8/2018 Lighthouse Report

2 Document uses legible font sizes

100% legible text ♥ ^

Font sizes less than 12px are too small to be legible and require mobile visitors to "pinch to zoom" in order to read. Strive to have >60% of page text ≥12px. Learn more.

Source	Selector	% of Page Text	Font Size
Legible text		100.00%	≥ 12px

Content Best Practices

Format your HTML in a way that enables crawlers to better understand your app's content.

Document has a <title> element

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. Learn more.

2 Links have descriptive text



Descriptive link text helps search engines understand your content. Learn more.

3 Document has a valid hreflang



hreflang links tell search engines what version of a page they should list in search results for a given language or region. Learn more.

Document avoids plugins



Search engines can't index plugin content, and many devices restrict plugins or don't support them. Learn more.

Crawling and Indexing

To appear in search results, crawlers need access to your app.

Page has successful HTTP status code



Pages with unsuccessful HTTP status codes may not be indexed properly. Learn more.

Page isn't blocked from indexing

Search engines are unable to include your pages in search results if they don't have permission to crawl them. Learn more.

Not applicable

1 audits ^

Crawling and Indexing

To appear in search results, crawlers need access to your app.

robots.txt is valid



If your robots txt file is malformed, crawlers may not be able to understand how you want your website to be crawled or indexed.

Runtime settings

• URL: http://127.0.0.1:8887/

- Fetch time: Aug 16, 2018, 10:50 AM GMT-5
- **Device**: Emulated Nexus 5X
- Network throttling: 150 ms TCP RTT, 1.638,4 Kbps throughput (Simulated)
- CPU throttling: 4x slowdown (Simulated)
- **User agent**: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari/537.36

Generated by **Lighthouse** 3.0.3 | File an issue