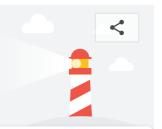
http://localhost:9000/

Jul 22, 2018, 9:40 AM GMT+1

Emulated Nexus 5X, Throttled Fast 3G network









Performance

Progressive Web App

Accessibility

Score scale:

0-44

45-74

75-100

Performance

Metrics

	95
- \	

First Contentful Paint	1,800 ms 🛇	First Meaningful Paint	3,030 ms 🔮
Speed Index	2,640 ms	First CPU Idle	3,030 ms 🗸
Time to Interactive	2,940 ms 🗸	Estimated Input Latency	17 ms 🔗

Values are estimated and may vary.





















Opportunities

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

Resource to optimize Estimated Savings

1 Enable text compression



0.6 s ^

Text-based responses should be served with compression (gzip, deflate or brotli) to minimize total network bytes. <u>Learn more</u>.

Uncompressed resource URL	Original	GZIP Savings
/js/dbhelper.js (localhost)	71 KB	52 KB
/css/styles.css (localhost)	12 KB	9 KB
/js/main.js (localhost)	9 KB	6 KB
/restaurants (localhost)	7 KB	5 KB
http://localhost:9000	3 KB	2 KB

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

URL	Original	Potential Savings
/js/dbhelper.js (localhost)	71 KB	54 KB
/js/main.js (localhost)	9 KB	4 KB

Avoid multiple, costly round trips to any origin

■ 0.15 s ^

Consider adding preconnect or dns-prefetch resource hints to establish early connections to important third-party origins. Learn more.

Origin	Potential Savings
http://localhost:1337	150 ms

Q Diagnostics

More information about the performance of your application.

Has significant main thread work

3,220 ms 🕕



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

Category	Time Spent
Script Evaluation	1,390 ms
Other	780 ms
Script Parsing & Compilation	440 ms
Style & Layout	176 ms
Rendering	167 ms
Garbage Collection	164 ms
Parse HTML & CSS	98 ms

2 JavaScript boot-up time

1,430 ms 🔮 🔨



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

URL	Total	Script Evaluation	Script Parsing & Compilation
/axe-adapter.js (Ihdoppojpmngadmnindnejefp	657 ms	269 ms	354 ms
/browser-sync/browser-sync-client.js?v=2	258 ms	257 ms	2 ms
dist/leaflet.js (unpkg.com)	129 ms	109 ms	3 ms
/js/main.js (localhost)	97 ms	82 ms	1 ms
/inject/index.js (eimadpbcbfnmbkopoojfekhnk	90 ms	60 ms	21 ms
js/inject.js (melpgahbngpgnbhhccnopmlmp	82 ms	69 ms	7 ms
:messaging ()	63 ms	63 ms	0 ms

URL	Total	Script Evaluation	Script Parsing & Compilation
/content.js (chklaanhfefbnpoihckbnefhakgoln	55 ms	40 ms	3 ms

3 Uses efficient cache policy on static assets

5 assets found <



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

URL	Cache TTL	Size (KB)
localhost:9000/c354368f-0c7d-4e98-a029-0a34b9828b53 ()	None	0 KB
1206/1540.jpg70?access_token=pk.eyJ1ljoidTs9mBSf98 (12 h	19 KB
1206/1539.jpg70?access_token=pk.eyJ1ljoidTs9mBSf98 (12 h	18 KB
1205/1539.jpg70?access_token=pk.eyJ1ljoidTs9mBSf98 (12 h	16 KB
1205/1540.jpg70?access_token=pk.eyJ1ljoidTs9mBSf98 (12 h	10 KB

Critical Request Chains

2 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

Longest chain: 1,769.6ms over 2 requests, totalling 38.1 KB

resources to improve page load. Learn more.

Initial Navigation

/ (localhost)

...dist/leaflet.css (unpkg.com) - 999.2ms, 0 KB

...dist/leaflet.js (unpkg.com) - 1,120.6ms, 38.1 KB

✓ Passed audits

15 audits ^

Eliminate render-blocking resources

1 resource delayed first paint by 0ms <



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

URL	Size (KB)	Download Time (ms)
dist/leaflet.css (unpkg.com)	0 KB	1,320 ms

2 Properly size images



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

3 Defer offscreen images



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

Minify CSS

Potential savings of 4 KB <



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

Potential **URL** Original Savings

/css/styles.css (localhost)

12 KB

4 KB

Potential savings of 9 KB <

0 ms

4/12

Potential savings of 0 ms <

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

Potential URL Original Savings /css/styles.css (localhost) 12 KB 9 KB

Efficiently encode images 6

Defer unused CSS

5

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.

Keep server response times low (TTFB) 8

Time To First Byte identifies the time at which your server sends a response. Learn more.

Avoid multiple page redirects 9

Redirects introduce additional delays before the page can be loaded. Learn more.

10 Preload key requests

Consider using <link rel=preload> to prioritize fetching late-discovered resources sooner. Learn more.

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

12 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
/js/dbhelper.js (localhost)	71 KB	390 ms
/images/6-821_large.jpg (localhost)	55 KB	310 ms
/images/8-821_large.jpg (localhost)	54 KB	300 ms
/images/2-821_large.jpg (localhost)	52 KB	290 ms
/browser-sync/browser-sync-client.js?v=2.24.5 (localhost)	44 KB	240 ms
/images/4-821_large.jpg (localhost)	43 KB	240 ms
/images/0-821_large.jpg (localhost)	40 KB	220 ms
/images/1-821_large.jpg (localhost)	40 KB	220 ms
/images/3-821_large.jpg (localhost)	39 KB	220 ms
dist/leaflet.js (unpkg.com)	38 KB	210 ms

13 Avoids an excessive DOM size

183



10



18

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.

```
<img alt="" role="presentation"</pre>
src="https://api.tiles.mapbox.com/v4/mapbox.streets/12/1205/1539.jpg70?
access token=pk.eyJ1IjoidC1tYWQiLCJhIjoiY2ppYWUycGx4MTNmYTN3bWxlMXFkZ2dpNiJ9.T
                                                                                   <head>
s9mBSf986SO2Y8hZCREqQ" class="leaflet-tile leaflet-tile-loaded" style="width:
256px; height: 256px; backface-visibility: hidden; transform:
translate3d(-98px, -24px, 0px); opacity: 1;">
```

14 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.

15 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.

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.

Additional items to manually check

3 audits ^

These checks are required by the baseline PWA Checklist 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.

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. Learn more.

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. Learn more.

Passed audits

11 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.

Responds 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. Learn more.

3 User can be prompted to Install the Web App

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

4 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.

5 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.

6 Registers 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. <u>Learn more</u>.

7 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>.

8 Configured for a custom splash screen



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

9 Address bar matches brand colors



The browser address bar can be themed to match your site. Learn more.

10 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. Learn more.

11 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.

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

<a href="http://leafletjs.com" title="A JS library for interactive
maps">Leaflet

OpenStreetMap

CC-BY-SA

Mapbox

<h2>Filter Results</h2>

Failing Elements

<button id="favorite-button-1" aria-label="Click to remove Mission Chinese Food
from your favorites!truefalse">*</button>

<h2>Mission Chinese Food</h2>

<a href="./restaurant.html?id=1" aria-label="Mission Chinese Food - View
Details">View Details

<button id="favorite-button-2" aria-label="Click to remove Emily from your
favorites!truefalse">*</button>

<h2>Emily</h2>

View
Details

<button id="favorite-button-3" aria-label="Click to remove Kang Ho Dong Baekjeong
from your favorites!truefalse">*</button>

<h2>Kang Ho Dong Baekjeong</h2>

<a href="./restaurant.html?id=3" aria-label="Kang Ho Dong Baekjeong - View
Details">View Details

<button id="favorite-button-4" aria-label="Click to add Katz's Delicatessen to
your favorites!falsetrue">*</button>

<h2>Katz's Delicatessen</h2>

<a href="./restaurant.html?id=4" aria-label="Katz's Delicatessen - View
Details">View Details

<button id="favorite-button-5" aria-label="Click to add Roberta's Pizza to your favorites!falsetrue">☆</button>

<h2>Roberta's Pizza</h2>

View
Details

<button id="favorite-button-6" aria-label="Click to add Hometown BBQ to your favorites!falsetrue">\alpha/button>

<h2>Hometown BBO</h2>

View
Details

<button id="favorite-button-7" aria-label="Click to add Superiority Burger to
your favorites!falsetrue">*</button>

<h2>Superiority Burger</h2>

<a href="./restaurant.html?id=7" aria-label="Superiority Burger - View
Details">View Details

<button id="favorite-button-8" aria-label="Click to add The Dutch to your favorites!falsetrue">☆</button>

<h2>The Dutch</h2>

Failing Elements

View
Details

<button id="favorite-button-9" aria-label="Click to add Mu Ramen to your favorites!falsetrue">></button>

<h2>Mu Ramen</h2>

View
Details

<button id="favorite-button-10" aria-label="Click to add Casa Enrique to your favorites!falsetrue">☆</button>

<h2>Casa Enrique</h2>

View
Details

<div>Copyright (c) 2018 </div>

<div> All Rights Reserved.</div>

Additional items to manually check

10 audits ^

These items address areas which an automated testing tool cannot cover. Learn more in our guide on <u>conducting an accessibility review</u>.

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.

3 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.

4 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.

^

9 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. <u>Learn more</u>.

✓ Passed audits

20 audits ^

Elements Use Attributes Correctly

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

1 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>.

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>.

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 [aria-*] attributes match their roles



Each ARIA `role` supports a specific subset of `aria-*` attributes. Mismatching these invalidates the `aria-*` attributes. Learn more.

2 [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.

3 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>.

4 [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. Learn more.

[role] values are valid



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

6 [aria-*] attributes have valid values



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

7 [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.

1 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. <u>Learn more</u>.

2 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. 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.

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.

2 Document has a <+ i+1e> 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.

Form elements have associated labels 3



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

Elements Are Well Structured

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

[id] attributes on the page are unique



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

2 Lists contain only elements and script supporting elements (<script> and <template>).



Screen readers have a specific way of announcing lists. Ensuring proper list structure aids screen reader output. Learn more.

List items () are contained within or parent elements



Screen readers require list items ('') to be contained within a parent '' or '' to be announced properly. Learn more.

Page Specifies Valid Language

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

<html> element has 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. Learn more.

2 <html> element has a valid value for its [lang] attribute



Specifying a valid BCP 47 language helps screen readers announce text properly. 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-scale] attribute is not less than 5.



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

14 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.

2 <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. <u>Learn more</u>.

3 <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>.

4 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>.

5 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. <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 <frame> or <iframe> elements have a title



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

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



A table being used for layout purposes should not include data elements, such as the thor caption elements or the summary attribute, because this can create a confusing experience for screen reader users. Learn more.

3 <object> elements have [alt] text



Screen readers cannot translate non-text content. Adding all text to `<object>` elements helps screen readers convey meaning to users. Learn more.

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



When a video provides a caption it is easier for deaf and hearing impaired users to access its information. <u>Learn</u> more.

5 <video> elements contain a <track> element with [kind="description"]



Audio descriptions provide relevant information for videos that dialogue cannot, such as facial expressions and scenes. <u>Learn more</u>.

Elements Are Well Structured



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

Ø .

When definition lists are not properly marked up, screen readers may produce confusing or inaccurate output. Learn more.

<dl>'s contain only properly-ordered <dt> and <dd> groups, <script> or <template> elements.

2 Definition list items are wrapped in <d1> elements



Definition list items (`<dt>` and `<dd>`) must be wrapped in a parent `<dl>` element to ensure that screen readers can properly announce them. Learn more.

Page Specifies Valid Language

11/12

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

1 [lang] attributes have a valid value



Specifying a valid <u>BCP 47 language</u> on elements helps ensure that text is pronounced correctly by a screen reader. <u>Learn more</u>.

Meta Tags Used Properly

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>.

Runtime settings

- URL: http://localhost:9000/
- Fetch time: Jul 22, 2018, 9:40 AM GMT+1
- Device: Emulated Nexus 5X
- Network throttling: 562.5 ms HTTP RTT, 1,474.6 Kbps down, 675 Kbps up (DevTools)
- CPU throttling: 4x slowdown (DevTools)
- User agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36

Generated by Lighthouse 3.0.3 I File an issue