

# Estimated Input Latency

## Overview

Input responsiveness is a key factor in how users perceive the performance of your app. Apps have 100ms to respond to user input. Any longer than that, and the user perceives the app as laggy. See [Measure Performance with the RAIL Model](#) for more information.

See [More information](#) for an explanation of why this audit tests for a target score of 50ms (rather than 100ms, which is what the RAIL model recommends).

## Recommendations

To make your app respond to user input faster, you need to optimize how your code runs in the browser. Check out the series of techniques outlined in the [Rendering Performance](#) docs. These tips range from offloading computation to web workers in order to free up the main thread, to refactoring your CSS selectors to perform less calculations, to using CSS properties that minimize the amount of browser-intensive operations.

One important caveat of this audit is that it's not a complete measurement of input latency. As explained in the [What this doc tests for](#) section of this doc, this audit does not measure how long your app truly takes to respond to a user input. In other words, it does not measure that your app's response to the user's input is visually complete.

To measure this manually, make a recording with the Chrome DevTools Timeline. See [How to use the Timeline Tool](#) for more help. The basic idea is to start a recording, perform the user input that you want to measure, stop the recording, and then analyze the flame chart to ensure that all stages of [the pixel pipeline](#) are complete within 50ms.

## More information

The RAIL performance model recommends that apps respond to user input within 100ms, whereas Lighthouse's target score is 50ms. Why?

The reason is that Lighthouse uses a proxy metric to measure how well your app responds to user input: availability of the main thread. Lighthouse assumes that your app needs 50ms to completely respond to the user's input (from performing any JavaScript executions

to physically painting the new pixels to the screen). If your main thread is unavailable for 50ms or more, that does not leave enough time for your app to complete the response.

There is a 90% probability a user would encounter input latency of the amount that Lighthouse reports, or less. 10% of users can expect additional latency.

The timing of this audit is from First Meaningful Paint to the end of the trace, which is roughly 5 seconds after the time to Consistently Interactive.

## Feedback

Was this page helpful?

Great! Thank you for the feedback.

Sorry to hear that. Please [open an issue](#) and tell us how we can improve.

---

*Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.*

*Last updated July 24, 2018.*