Are long JavaScript tasks delaying your Time to Interactive?

- What are Long Tasks?
- Are there Long Tasks in my page that could delay interactivity?
- What is causing my Long Tasks?
- What are common ways to optimize Long Tasks?

Long Tasks can keep the main thread busy, delaying user interaction. Chrome DevTools can now visualize Long Tasks, making it easier to see tasks to optimize.

If you use Lighthouse to audit your pages, you may be familiar with Time to Interactive, a metric representing when users can interact with your page and get a response. But did you know Long (JavaScript) Tasks can contribute heavily to a poor TTI?

Time to Interactive displayed in the Lighthouse Report

What are Long Tasks?

A Long Task is JavaScript code that monopolizes the main thread for extended periods of time, causing the UI to "freeze".

While a web page is loading, Long Tasks can tie up the main thread and make the page unresponsive to user input even if it looks ready. Clicks and taps often don't work because event listeners, click handlers etc have not yet been attached.

CPU-heavy Long Tasks occur due to complex work that takes longer than 50ms. Why 50ms? The RAIL model suggests you process user input events in 50ms to ensure a visible response within 100ms. If you don't, the connection between action and reaction is broken.

While the RAIL model suggests providing a visual response to user input within 100 ms, the Interaction to Next Paint (INP) metric's thresholds allow for up to 200 ms in order to set more easily achievable expectations, especially for slower devices.

Are there Long Tasks in my page that could delay interactivity?

Until now, you've needed to manually look for "long yellow blocks" of script over 50ms long in Chrome DevTools or use the Long Tasks API to figure out what tasks were delaying interactivity. This could be a little cumbersome.

A DevTools Performance panel screenshot showing the differences between short tasks and long tasks

To help ease your performance auditing workflow, DevTools now visualizes Long Tasks. Tasks (shown in gray) have red flags if they are Long Tasks.

DevTools visualizing Long Tasks as gray bars in the Performance Panel with a red flag for long tasks

- Record a trace in the Performance panel of loading up a web page.
- Look for a red flag in the main thread view. You should see tasks are now gray ("Task").
- Hovering over a bar will let you know the duration of the task and if it was considered "long".

What is causing my Long Tasks?

To discover what is causing a long task, select the gray **Task** bar. In the drawer beneath, select **Bottom-Up** and **Group by Activity**. This allows you to see what activities contributed the most (in total) to the task taking so long to complete. Below, it appears to be a costly set of DOM queries.

Selecting a long task (labelled 'Task') in DevTools allows us to drill-down into the activities that were responsible for it.

What are common ways to optimize Long Tasks?

Large scripts are often a major cause of Long Tasks so consider splitting them up. Also keep an eye on third-party scripts; their Long Tasks can delay primary content from getting interactive.

Break all your work into small chunks (that run in < 50ms) and run these chunks at the right place and time; the right place may even be off the main thread, in a worker. Phil Walton's Idle Until Urgent is a good read on this topic. See also the optimize long tasks article for general strategies for managing and breaking up long tasks.

Keep your pages responsive. Minimizing Long Tasks is a great way to ensure your users have a delightful experience when they visit your site. For more on Long Tasks, check out User-centric Performance Metrics.

□□□ #3
□ □□□ □□□ 2026-10-31 22:01:57 UTC
□ cocobook-sync □□□ 2026-11-04 17:37:27 UTC