With experimental “Never slow mode,” Chrome tries to stop Web devs making it slow



Enlarge / Google wants less of this.

reader comments

104 with 82 posters participating

Share this story

Since Chrome’s very first release, performance has been one of Google’s top priorities. But Google is against a competing force: Web developers. The Web of today is a more-complex, bandwidth-intensive place than it was when Chrome was first released, which means that—although Internet connections and the browser itself are faster than they’ve ever been—slow pages remain an everyday occurrence.

Google engineers have been developing “Never Slow Mode” in a bid to counter this. Spotted at Chrome Story (via ZDNet), the new mode places tight limitations on Web content in an effort to make its performance more robust and predictable.

The exact design and rationale of Never Slow Mode aren’t public—the changelog for the feature mentions a design document but says it’s currently Google-internal. But taken together, that design and rationale will ensure that the browser’s main thread never has to do too much work and will never get too delayed. They will also ensure that only limited amounts of data are pulled down over the network. This should make the browser more responsive to user input, lighter on the network, and a bit less of a memory hog than it would otherwise be.

As well as capping various data sizes and limiting the amount of time that JavaScripts can run, Never Slow Mode also blocks access to certain features that webpages can currently use that incur a performance cost. In particular, scripts are prohibited from using document.write(), which is widely used by scripts to dynamically emit HTML (potentially including embedded CSS and JavaScript) into a page. They’re also blocked from making synchronous XMLHttpRequests to transfer data to and from servers. Synchronous requests tend to make pages feel slow, because the browser can’t run any other scripting while it’s waiting for the synchronous request to complete. Asynchronous XMLHttpRequests remain supported, as these let the browser do other things while it’s waiting for the remote server to respond.

The budgets for execution resources get reset each time a user interacts with the page. So each time a page gets scrolled or tapped, it can run a bit more JavaScript and pull down a bit more data.

The size and execution-time limitations are draconian, to say the least, and the JavaScript changes will outright break many

existing pages. Together, they make Never Slow Mode a little mysterious: this isn’t a mode that can be used for general purpose Web browsing, because pages will either run out of resources or depend on forbidden JavaScript. The current implementation of Never Slow Mode notes that it’s only a prototype (with an “idiotic implementation,” no less). So whatever its ultimate purpose, there’s still much development left.

We’ve asked Google for comment but have heard nothing at the time of writing.

https://arstechnica.com/?p=1451707