

Background Tabs in Chrome 57

Background tabs can have a dramatic negative effect on browser performance, especially on battery life. To mitigate this, Chrome has been placing various restrictions on background tabs for the last several years. Recently there's been a number of efforts to make further improvements, and this document gives an overview of the Chrome policy. This document focuses on describing current policies in Chrome 57. Long-term strategy and further plans can be found in [this document](#).

Optimising an application for background

Web developers should be aware that users often have a lot of tabs open in the background and it can have a serious effect on power usage and battery life. Work in the background should be kept to a minimum unless it's absolutely necessary to provide a particular user experience. The [Page visibility API](#) should be used to detect when page is the backgrounded and suspend all unnecessary work like visual updates.

For some sites, this simple optimisation can reduce CPU usage by as much as 75%:

```
var doVisualUpdates = true;

document.addEventListener('visibilitychange', function(){
  doVisualUpdates = !document.hidden;
});

function update() {
  if (!doVisualUpdates) {
    return;
  }
  doStuff();
}
```



Policies

requestAnimationFrame()

Per [the documentation](#), Chrome does not call `requestAnimationFrame()` when a page is in the background. This behavior has been in place since 2011.

Background timer alignment

Since [Chrome 11](#), each independent timer is run no more than once per second. Chrome runs these timers in batches once a second, ensuring that the number of process wakeups is kept to a minimum. Pages playing audible audio are considered user-visible and exempted from background timer throttling. Exemption lasts for several seconds after audio stops playing to allow applications to queue the next audio track.

Note that audio is considered audible when and only when Chrome shows the audio icon. Silent audio streams do not grant exemptions.

Budget-based background timer throttling

[Shipping in Chrome 57](#), budget-based timer throttling is a further extension of the timer alignment mechanism, placing an additional limit on background timer CPU usage. It operates as follows:

- Each background tab has a time budget (in seconds) for running timers in the background.
- A page is subjected to time budget limitations after 10 seconds in the background.
- A timer task is allowed to run only when the time budget is non-negative.
- After a timer has executed, its run time is subtracted from the budget.
- The budget continuously regenerates with time (currently set to a rate of 0.01 seconds per second). Note that this budget regeneration rate can be tweaked as Chrome collects more data about throttling behavior.

There are a number of automatic exemptions from this throttling:

- Applications playing audio are considered foreground and aren't throttled.
- Applications with real-time connections (WebSockets and WebRTC), to avoid closing these connections by timeout. The run-timers-once-a-second rule is still applied in these cases.

Note that this mechanism uses wall time, not CPU time. It's a good approximation of CPU time and penalises blocking the main thread for a long time.

Finally, remember that if you are using long tasks in the background, your application can be throttled for a very long period of time (up to 100 times the duration of your task). Split your work in to chunks of 50ms or less per [the performance guidelines](#) and use the `visibilityChange` listener to avoid doing unnecessary work in background.

Opt-outs

Chrome provides the `--disable-background-timer-throttling` flag for use cases like running test suites and other user-sanctioned heavy computations.

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 March 14, 2017.