

# Deprecations and removals in Chrome 64

To see all deprecations and removals for this and previous versions of Chrome, visit the [deprecations page](#).



**By Joseph Medley**

Technical Writer

In nearly every version of Chrome, we see a significant number of updates and improvements to the product, its performance, and also capabilities of the Web Platform. This article describes some of the deprecations and removals in Chrome 64, which is in beta as of December 14.

## Remove support for multiple shadow roots

Shadow DOM version 0 allowed multiple shadow roots. At a [standards meeting in April 2015](#) it was decided that this feature should not be part of version 1. Support was deprecated shortly thereafter in Chrome 45. In Chrome 64 support is now removed.

[Intent to Remove](#) | [Chromestatus Tracker](#) | [Chromium Bug](#)

## Remove `getMatchedCSSRules()`

The `getMatchedCSSRules()` method is a non-standard, WebKit-only API that retrieves a list of style rules applied to a particular element. This has been [deprecated since 2014](#). It's now being removed because it's not on a standards track.

Since there is currently no standards-based alternative, developers would need to create their own. There is at least [one example on StackOverflow](#).

[Intent to Remove](#) | [Chromestatus Tracker](#) | [Chromium Bug](#)

## Deprecation policy

To keep the platform healthy, we sometimes remove APIs from the Web Platform which have run their course. There can be many reasons why we would remove an API, such as:

- They are superseded by newer APIs.
- They are updated to reflect changes to specifications to bring alignment and consistency with other browsers.
- They are early experiments that never came to fruition in other browsers and thus can increase the burden of support for web developers.

Some of these changes will have an effect on a very small number of sites. To mitigate issues ahead of time, we try to give developers advanced notice so they can make the required changes to keep their sites running.

Chrome currently has a [process for deprecations and removals of API's](#), essentially:

- Announce on the [blink-dev](#) mailing list.
- Set warnings and give time scales in the Chrome DevTools Console when usage is detected on the page.
- Wait, monitor, and then remove the feature as usage drops.

You can find a list of all deprecated features on [chromestatus.com](#) using the [deprecated filter](#) and removed features by applying the [removed filter](#). We will also try to summarize some of the changes, reasoning, and migration paths in these posts.

---

*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 2, 2018.*