Deprecations and removals in Chrome 66

To see all deprecations and removals for this and previous versions of Chrome, visit the deprecations page.

By Joseph Medley
Technical Writer

ImageCapture.setOptions() removed

Current thinking on setting device options is to use the constrainable pattern . Consequently this property was removed from the ImageCapture specification . Since this method appears to have little to no use on production websites, it is being removed. A replacement method is not available at this time.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Service worker: disallow CORS responses for same-origin requests

Previous versions of the service worker specification allowed a service worker to return a CORS response to a same-origin request. The thinking was that the service worker could read from a CORS response to create a completely synthetic response. In spite of this, the original request URL was maintained in the response. So `outerResponse.url` exactly equaled `url` and `innerResponse.url` exactly equaled `crossOriginURL`.

A recent change to the Fetch specification requires that `Response.url` be exposed if it is present. A consequence of this is scenarios in which `self.location.href` returns a different origin than `self.origin`. To avoid this, service workers are no longer allowed to return CORS responses for same origin requests.

For a longer discussion on this change, see the issue filed against the Fetch specification in November 2017.

Chromestatus Tracker | Chromium Bug

WebAudio: dezippering removed

Web audio originally shipped with dezippering support. When an AudioParam value was set directly with the value setter, the value was not updated immediately. Instead, an exponential smoother was applied with a time constant of about 10 ms so that the change was done smoothly, limiting glitches. It was never specified which parameters had smoothing and what the time constant was. It wasn’t even obvious if the actual time constant was the appropriate value.

After much discussion , the working group removed dezippering from the spec. Now, the value is changed immediately when set. In place of dezippering, it is recommended that developers use the existing `AudioParam.setTargetAtTime()` method to do the dezippering, giving you full control on when to apply it, how fast to change, and on which parameters should be smoothed.

Removing this reduces developer confusion which audio parameters support dezippering.

Intent to Remove | Chromestatus Tracker | Chromium Bug

CSS position values with three parts deprecated

Recently specifications have required that new properties accepting position values not support values with three parts. It's believed this approach makes processing shorthand syntax easier. The current version of the CSS Values and Units Module applies this requirement to all CSS position values. As of Chrome 66, three-part position values are deprecated. Removal is expected in Chrome 68, around July 2018.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Methods document.createTouch(), document.createTouchList() are deprecated

The `TouchEvent()` constructor has been supported in Chrome since version 48. To comply with the specification, `document.createTouch()` and `document.createTouchList()` are now deprecated.

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.

Last updated July 2, 2018.