API Deprecations and Removals in Chrome 52

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 the changes in Chrome 52, which is in beta as of June 9. This list is subject to change at any time.

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- They are superseded by newer APIs.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.

Deprecate MediaStream ended event and attribute and onended attribute

TL;DR: The **ended** event and attribute and the **onended** event handler are being deprecated because they have been removed from the Media Capture and Streams spec.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Neither the **ended** event, nor the **onended** event handler have been part of the WebRTC spec for about three years. Developers wanting to watch events should use `MediaStreamTracks` instead of `MediaStreams`.

Removal is anticipated in Chrome 53.

Block pop-ups from cross-origin iframes during touch events except during a tap gesture

TL;DR: Chrome will begin disallowing pop-ups and other sensitive operations on touch events that don't correspond to a tap from inside of cross-origin iframes.

Intent to Remove | Chromestatus Tracker | Chromium Bug

By their very nature, touch events can be ambiguous when compared to their corresponding mouse events. For example, if a user slides a finger across the screen, is said user sliding a toggle switch or scrolling the view? Some third-party content in iframes have taken advantage of this ambiguity to intentionally disable scrolling on the containing page.

To combat this, pop-ups and other sensitive operations will be disallowed on touch events from cross-origin iframes. The touchend event will continue to behave as before.

Deprecate overload of postMessage()

TL;DR: An unneeded and little-used variant of the `postMessage()` interface is being deprecated, specifically `postMessage(message, transferables, targetOrigin)`.

Intent to Remove | Chromestatus Tracker | Chromium Bug

The `postMessage()` method is a way to securely communicate between the scripts of pages on different origins. WebKit/Blink supports three versions:

- `postMessage(message, targetOrigin)`
- `postMessage(message, targetOrigin, transferables)`
- `postMessage(message, transferables, targetOrigin)`

The last item in this list was an accident from the history of the spec's evolution and implementation. Because it is rarely used, it will be deprecated and later removed. This applies to both `window.postMessage()` and `worker.postMessage()`.

Removal is anticipated in Chrome 54.

Remove support for X-Frame-Options in tags

TL;DR: To both comply with the spec and increase consistency with other browsers, support for `X-Frame-Options` inside a `<meta>` tag is being removed.

Intent to Remove | Chromium Bug

The `X-Frame-Options` HTTP response header indicates whether a browser can render a page in an`<frame>`, `<iframe>`, or `<object>` tag. This lets a site avoid clickjacking since such pages cannot be embedded in other sites. The current version of the X-Frame-Options spec explicitly restricts user agents from supporting this field inside a `<meta>` tag.

To both comply with the spec and increase consistency with other browsers, support for `X-Frame-Options` inside a `<meta>` tag is being removed.

Remove non-primary button click event

TL;DR: Non-primary mouse clicks no longer fire click events, but `MouseEvent.button` is still available.

Intent to Remove | Chromestatus Tracker | Chromium Bug

To bring Chrome in line with the UIEvents spec, we're removing the click events for non-primary mouse buttons. Non-primary mouse buttons vary by device. Generally this means anything other than a right or left mouse button. Note that the precise button clicked may still be retrieved using the `MouseEvent.button` property sent to events such as `mousedown` or `mouseup`.

Remove requestAutocomplete()

The `requestAutocomplete()` function allowed forms to be filled out on demand by the browser's autofill capability. Yet more than two years in, this capability is only supported in Blink and its usage is low. For these reasons, `requestAutocomplete()` is removed in Chrome 52.

Intent to Remove

Last updated July 2, 2018.