API Deprecations and Removals in Chrome 57

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 deprecations and removals in Chrome 57, which is in beta as of early February. This list is subject to change at any time.

Remove BluetoothDevice.uuids attribute

The `BluetoothDevice.uuids` attribute is being removed to bring the Web Bluetooth API in line with the current specification. You can retrieve all allowed GATT services by calling device.getPrimaryServices().

Chromium Bug

Remove key generation element

Since Chrome 49, `<keygen>`'s default behaviour has been to return the empty string, unless a permission was granted to this page. IE/Edge do not support `<keygen>` and have not indicated public signals to support `<keygen>`. Firefox already gates `<keygen>` behind a user gesture, but is publicly supportive of removing it. Safari ships `<keygen>` and has not expressed public views regarding its continued support. With Chrome 57, this element is removed.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Remove prefixed resource timing buffer-management API

Two methods and an event handler, `webkitClearResourceTimings()`, `webkitSetResourceTimingBufferSize()`, and `onwebkitresourcetimingbufferfull` are obsolete and vendor-specific. The standard versions of these APIs have been supported in since Chrome 46, and the prefixed functions were deprecated in that version as well. These features were originally implemented in WebKit, but Safari has not enabled them. Firefox, IE 10+, and Edge have only unprefixed version of the API. Therefore the webkit versions are being removed.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Remove ServiceWorkerMessageEvent in favor of using MessageEvent

The HTML spec extended `MessageEvent` to allow `ServiceWorker` as a type for the `source` attribute. `client.postMessage()` and creation of custom message events are changed to use `MessageEvent` instead of `ServiceWorkerMessageEvent`. `ServiceWorkerMessageEvent` is removed.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Remove webkit-prefixed IndexedDB global aliases

The `IndexedDB` entry point and global constructors were exposed with `webkit` prefixes somewhere around Chrome 11. The non-prefixed versions were added in Chrome 24 and the prefixed versions were deprecated in Chrome 38. The following interfaces are affected:

- `webkitIndexedDB` (main entry point)
- `webkitIDBKeyRange` (non-callable global constructor, but has useful static methods)
- `webkitIDBCursor`
- `webkitIDBDatabase`
- `webkitIDBFactory`
- `webkitIDBIndex`
- `webkitIDBObjectStore`
- `webkitIDBRequest`
- `webkitIDBTransaction` (non-callable global constructors)

Intent to Remove | Chromestatus Tracker | Chromium Bug

WebAudio: Remove prefixed AudioContext and OfflineAudioContext

Chrome has supported `WebAudio` since mid 2011, including `AudioContext`. `OfflineAudioContext` was added the following year. Given how long the standard interfaces have been supported and Google's long-term goal of removing prefixed features, the prefixed versions of these interfaces have been deprecated since late 2014 and are now being removed.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Deprecate and remove webkitCancelRequestAnimationFrame

The `webkitCancelRequestAnimationFrame()` method is a an obsolete, vendor-specific API and the standard `cancelAnimationFrame()` has long been supported in Chromium. Therefore the webkit version is being removed.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Deprecate case-insensitive matching for usemap attribute

The `usemap` attribute was formerly defined as caseless. Unfortunately implementing this was complicated enough that no browsers implemented it correctly. Research suggested that such complicated algorithm is unnecessary, and even ASCII case-insensitive matching is unnecessary.

Consequently, the specification was updated so that case-sensitive matching is applied. The old behavior is deprecated in Chrome 57, with removal expected in Chrome 58.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Deprecate FileReaderSync in service workers

The Service Worker spec has always had the (non-normative) note that "any type of synchronous requests must not be initiated inside of a service worker", to avoid blocking the service worker. Blocking the service worker would block all network requests from controlled pages. Unfortunately, the `FileReaderSync` API has long been available in service workers.

Currently only Firefox and Chrome expose `FileReaderSync` in service workers. There's agreement from Firefox in the spec discussion that this should be fixed. Removal is anticipated in Chrome 59.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Deprecate legacy caller for HTMLEmbedElement and HTMLObjectElement

That an interface has a legacy caller means that an instance can be called as a function. Currently, `HTMLEmbedElement` and `HTMLObjectElement` support this functionality. In Chrome 57 this ability is deprecated. After removal, which is expected in Chrome 58, calling will throw an exception.

This change brings Chrome in line with recent spec changes. The legacy behavior is not supported in Edge or Safari, and it is being removed from Firefox.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Deprecate RTCRtcpMuxPolicy of "negotiate"

The `rtcpMuxPolicy` is used by Chrome to specify its preferred policy regarding use of RTP/RTCP multiplexing. In Chrome 57, we changed the default `rtcpMuxPolicy` to "require" and deprecated "negotiate" for following reasons:

- Non-muxed RTCP uses extra network resources.
- Removing "negotiate" will make the API surface simpler, since an "RtpSender"/"RtpReceiver" will then only ever have a single transport.

In Chrome 57, "negotiate" is deprecated. We believe this is a non-breaking change since the user will get a deprecation message and `RTCPeerConnection` can still be created. Removal is in Chrome 63.

Intent to Deprecate | Chromium Bug

Deprecate support for embedded credentials in subresource requests

Hard-coding credentials into subresource requests is problematic from a security perspective, as it's allowed hackers to brute-force credentials in the past. These dangers are exacerbated for credentialed subresource requests that reach into internal IP ranges (your routers, etc.). Given the low usage, closing this (small) security hole seems quite reasonable.

Developers can embed resources that do not require basic/digest auth, relying instead on cookies and other session management mechanisms.

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.