API Deprecations and Removals in Chrome 53

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- They are superseded by newer APIs.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.

DHE-based ciphers being phased out

TL;DR: DHE-based ciphers are removed in Chrome 53, desktop because they're insufficient for long term use. Servers should employ ECDHE, if it's available or a plain-RSA cipher if it is not.

Intent to Remove | Chromestatus Tracker | Chromium Bug

Last year, we Chrome the minimum TLS Diffie-Hellman group size from 512-bit to 1024-bit; however, 1024-bit is insufficient for the long-term. Metrics report that around 95% of DHE connections seen by Chrome use 1024-bit DHE. This, compounded with how DHE is negotiated in TLS, makes it difficult to move past 1024-bit.

Although there is a draft specification that fixes this problem, it is still a draft and requires both client and server changes. Meanwhile, ECDHE is already widely implemented and deployed. Servers should upgrade to ECDHE if available. Otherwise, ensure a plain-RSA cipher suite is enabled.

DHE-based ciphers have been deprecated since Chrome 51. Support is being removed from desktop in Chrome 53.

FileError deprecation warning

TL;DR: Removal of the deprecated `FileError` interface is expected in Chrome 54. Replace references to **`err.code`** with **`err.name`** and **`err.message`**.

Intent to Remove | Chromestatus Tracker | Chromium Bug

The current version of the File API standard does not contain the `FileError` interface and it's support was deprecated some time in 2013. In Chrome 53, this deprecation warning will be printed to the DevTools console:

'FileError' is deprecated and will be removed in 54. Please use the 'name' or 'message' attributes of the error rather than 'code'.

This has different effects in different contexts.

- `FileReader.error` and `FileWriter.error` will be `DOMException` objects instead of `FileError` objects.
- For *asynchronous* `FileSystem` calls the `ErrorCallback` will be passed `FileError.ErrorCode` instead of `FileError`.

- For *synchronous* `FileSystem` calls `FileError.ErrorCode` will be thrown instead of `FileError`.

This change only impacts code that relies on comparing the error instance's code (`e.code`) directly against `FileError` enum values (`FileError.NOT_FOUND_ERR`, etc). Code that tests against hard coded constants (for example `e.code === 1`) could fail by reporting incorrect errors to the user.

Fortunately the `FileError`, `DOMError`, and `DOMException` error types all share `name` and `message` properties which give consistent names for error cases (in other words, `e.name === "NotFoundError"`). Code should use those properties instead, which will work across browsers and continue to work once the `FileError` interface itself has been removed.

The removal of `FileError` is anticipated Chrome 54.

Remove results attribute for <input type=search>

TL;DR: The `results` attribute is being removed because it's not part of any standard and is inconsistently implemented across browsers.

Intent to Remove | Chromestatus Tracker | Chromium Bug

The `results` value is only implemented in webkit and behaves highly inconsistently on those that do. For example, Chrome adds a magnifier icon to the input box, while on Safari desktop, it controls how many previous searches are shown in a popup shown by clicking the magnifier icon. Since this isn't part of any standard, it's being deprecated.

If you still need to include the search icon in your input field then you will have to add some custom styling to the element. You can do this by including a background image and specifying a left padding on the input field.

```
input[type=search] {
  background: url(some-great-icon.png) no-repeat scroll 15px 15px;
  padding-left:30px;
}
```

This attribute has been deprecated since Chrome 51.

Last updated July 2, 2018.