Audio/Video Updates in Chrome 58

By François Beaufort

Dives into Chromium source code

- Developers can now customize media controls such as the download, fullscreen and remoteplayback buttons.
- Sites installed using the "Add to Homescreen" flow can autoplay audio and video in the manifest's scope.
- Chrome on Android now pauses autoplaying a muted video when it is invisible.
- Developers can now access the approximate range of colors supported by Chrome and output devices using the `color-gamut` Media Query.
- When using Media Source Extensions, it's now possible to switch between encrypted and clear streams.

Media controls customization

Developers can now customize Chrome's native media controls such as the download, fullscreen and remoteplayback buttons using the new ControlsList API.

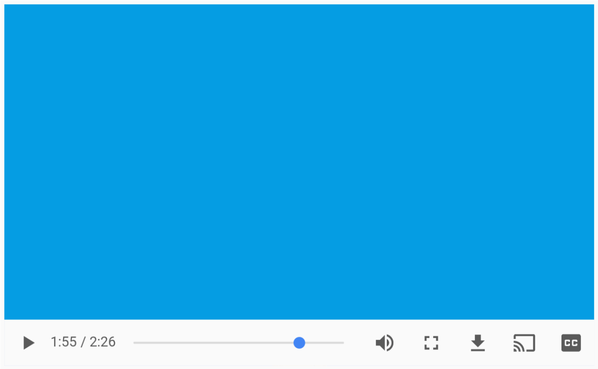

Figure 1. Native media controls in Chrome 58

This API offers a way to show or hide native media controls that do not make sense or are not part of the expected user experience, or only whitelist a limited set of features.

The current implementation for now is a blacklist mechanism on native controls with the ability to set them directly from HTML content using the new attribute `controlsList`. Check out the official sample.

Usage in HTML:

```
<video controls controlsList="nofullscreen nodownload noremoteplayback"></vide
```

Usage in JavaScript:

```
var video = document.querySelector('video');
video.controls; // true
video.controlsList; // ["nofullscreen", "nodownload", "noremoteplayback"]
video.controlsList.remove('noremoteplayback');
video.controlsList; // ["nofullscreen", "nodownload"]
video.getAttribute('controlsList'); // "nofullscreen nodownload"

video.controlsList.supports('foo'); // false
video.controlsList.supports('noremoteplayback'); // true
```

Intent to Ship | Chromestatus Tracker | Chromium Bug

Autoplay for Progressive Web Apps added to home screen

Previously, Chrome used to block all `autoplay` with sound on Android without exception. This is no longer true. From now on, sites installed using the improved Add to Home Screen flow are allowed to autoplay audio and video served from origins included in the web app manifest's scope without restrictions.

```
{
  "name": "My Web App",
  "description": "An awesome app",
  "scope": "/foo",
  ...
}
```

```
<html>
  <link rel="canonical" href="https://example.com/foo">
  <audio autoplay src="https://cdn.com/file.mp4"></audio>
</html>
```

Audio will autoplay as `/foo` is in the scope.

```
<html>
  <link rel="canonical" href="https://example.com/bar">
  <audio autoplay src="https://cdn.com/file.mp4"></audio>
</html>
```

Audio fails to autoplay as `/bar` is NOT in the scope.

Intent to Ship | Chromestatus Tracker | Chromium Bug

Pause autoplaying muted video when invisible

As you may already know, Chrome on Android allows `muted` videos to begin playing without user interaction. If a video is marked as `muted` and has the `autoplay` attribute, Chrome starts playing the video when it becomes visible to the user.

From Chrome 58, in order to reduce power usage, playback of videos with the `autoplay` attribute will be paused when offscreen and resumed when back in view, following Safari iOS behavior.'

Note: This only applies to videos that are declared as `autoplay` but not videos that start playing with `play()`.

Intent to Ship | Chromestatus Tracker | Chromium Bug

color-gamut media query

As wide color gamut screens are more and more popular, sites can now access the approximate range of colors supported by Chrome and output devices using the `color-gamut` media query.

If you're not familiar yet with the definitions of color space, color profile, gamut, wide-gamut and color depth, I highly recommend you read the Improving Color on the Web WebKit blog post. It goes into much detail on how to use the `color-gamut` media query to serve wide-gamut images when the user is on wide-gamut displays and fallback to sRGB images otherwise.

The current implementation in Chrome accepts the `srgb`, `p3` (gamut specified by the DCI P3 Color Space), and `rec2020` (gamut specified by the ITU-R Recommendation BT.2020 Color Space) keywords. Check out the official sample.

Usage in HTML:

```
<picture>
  <source media="(color-gamut: p3)" srcset="photo-p3.jpg">
  <source media="(color-gamut: rec2020)" srcset="photo-rec2020.jpg">
  <img src="photo-srgb.jpg">
</picture>
```

Usage in CSS:

```
main {
  background-image: url("photo-srgb.jpg");
}

@media (color-gamut: p3) {
  main {
    background-image: url("photo-p3.jpg");
  }
}

@media (color-gamut: rec2020) {
  main {
    background-image: url("photo-rec2020.jpg");
  }
}
```

Usage in JavaScript:

```
// It is expected that the majority of color displays will return true.
if (window.matchMedia("(color-gamut: srgb)").matches) {
  document.querySelector('main').style.backgroundImage = 'url("photo-srgb.jpg")';
}

if (window.matchMedia("(color-gamut: p3)").matches) {
  document.querySelector('main').style.backgroundImage = 'url("photo-p3.jpg")';
}

if (window.matchMedia("(color-gamut: rec2020)").matches) {
  document.querySelector('main').style.backgroundImage = 'url("photo-rec2020.jpg")';
}
```

For info, this screen currently supports approximately:

✔ the sRGB gamut or more.

✕ the gamut specified by the DCI P3 Color Space or more.

✕ the gamut specified by the ITU-R Recommendation BT.2020 Color Space or more.

Intent to Ship | Chromestatus Tracker | Chromium Bug

Last updated July 2, 2018.