New in Chrome 61

By Pete LePage

Pete is a Developer Advocate

- Chrome 61 now supports JavaScript modules natively, unifying the way modular JavaScript can be written.

- You can now use `navigator.share` to trigger the native Android share dialog.

- The WebUSB API has landed, allowing web apps to access user permitted USB devices.

- And there's plenty more!

Note: Want the full list of changes? Check out the Chromium source repository change list.

I'm Pete LePage. Let's dive in and see what's new for developers in Chrome 61!

JavaScript Modules

Chrome 61 adds native support for JavaScript modules via the `<script type="module">` element. That makes it possible for Chrome to fetch granular dependencies in parallel, taking advantage of caching, avoiding duplications across the page and ensuring that script executes in the correct order.

```
<script type="module">
  import {addText} from './utils.js';
```

```
    addText('Modules are pretty cool.');
</script>
```
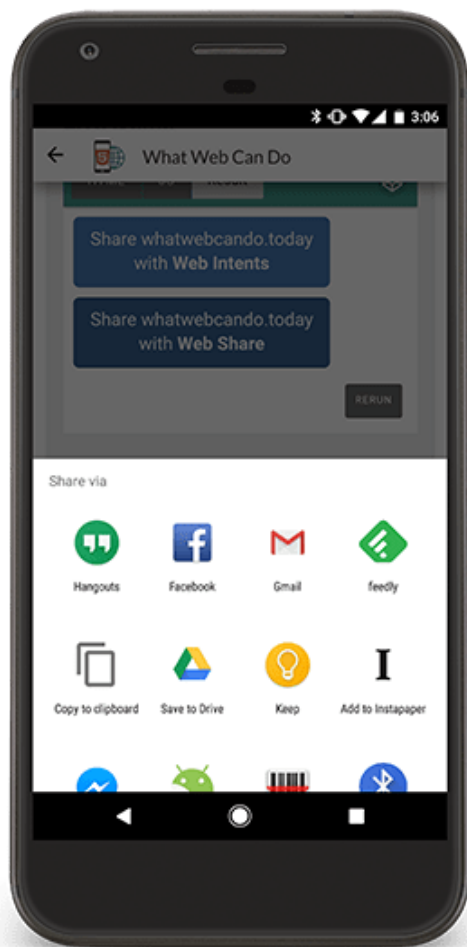
This standardized module system unifies the way modular JavaScript can be written and shipped to web browsers. In the future, the same system will be available in Node, making it easier for you to write and deploy isomorphic JavaScript.

You can learn more about modules and the aspects of JavaScript that are affected by modules from the links below.

- Chrome Status
- ES Modules in Browsers
- ES6 Modules in Depth

Web Share API

If you want users to be easily able to share your content on their favorite social network, you need to integrate sharing buttons into your site for each social network. It adds bloat to your page, doesn't always fit your UI nicely, and means you need to include code from a third party site.

The Web Share API, available today on Chrome for Android allows you to invoke the native sharing capabilities of the users device, allowing the user to easily share text or links with any of their installed native apps!

In a future release, this API will also be able to share to installed web apps. To use it, simply call `navigator.share` with the details of the page you want to share the system will handle the rest.
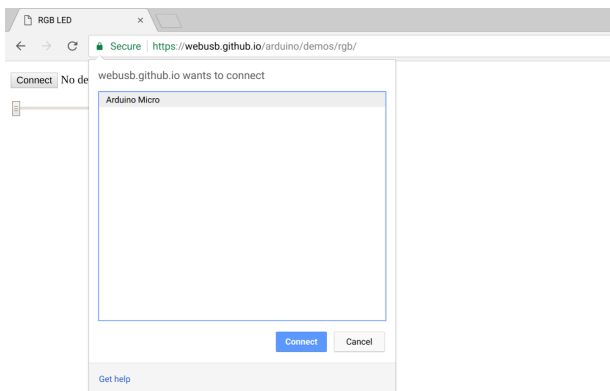
```
navigator.share({
  title: document.title, text: 'Hello',
  url: window.location.href
}).then(() => {
```

```
  console.log('Successful share');
});
```

Check out Paul's <u>WebShare API Update</u> for full details and some best practices that you should be following.

WebUSB

Most hardware peripherals such as keyboards, mice, printers, and gamepads are supported by high-level web platform APIs. But, using specialized educational, scientific, industrial or other USB devices in the browser has been hard, often requiring specialized drivers.

Chrome now supports the WebUSB API, allowing web apps to communicate with USB devices, after the user has provided their consent. To learn more about the security and privacy considerations and how they're addressed, have a peek at the <u>WebUSB spec</u>.

Then, when you're ready to dive in, take a look at Francois' <u>WebUSB post</u> on updates.

And more!

- You can now specify scrolling smoothness with the `scroll-behavior` CSS property.
- <u>CSS hex color values</u> can now specify alpha transparency by adding digits to the end of the string.
- You can access the relative positions of the screen content with the <u>Visual Viewport API</u>, exposing complex functionality like pinch-and-zoom in a more direct way.

These are just a few of the changes in Chrome 61 for developers.

Then subscribe to our YouTube channel, and you'll get an email notification whenever we launch a new video, or add our RSS feed to your feed reader.

I'm Pete LePage, and as soon as Chrome 62 is released, I'll be right here to tell you – what's new in Chrome!

Last updated July 2, 2018.