Introducing the Web Share API

By Paul Kinlan

Paul is a Developer Advocate

By Sam Thorogood

Evangelises Chrome and the mobile web in the Developer Relations team at Google.

Good news, everybody! In Chrome 61 for Android, we've launched the `navigator.share()` method, which allows websites to invoke the native sharing capabilities of the host platform.

This method, part of the simple Web Share API—written by Matt Giuca on the Chrome team—allows you easily trigger the native Android share dialog, passing either a URL or text to share. This is an important API as it gives your end-users user control of how and where the data is shared.

Usage

The Web Share API is a Promise-based, single method API. It accepts an object which must have at least one of the properties named `text` or `url`.

```
if (navigator.share) {
  navigator.share({
      title: 'Web Fundamentals',
      text: 'Check out Web Fundamentals - it rocks!',
      url: 'https://developers.google.com/web',
  })
    .then(() => console.log('Successful share'))
    .catch((error) => console.log('Error sharing', error));
}
```

Once invoked it will bring up the native picker (see video) and allow you to share the data with the app chosen by the user.

To use the Web Share API:

- you must be served over HTTPS
- you can only invoke the API in response to a user action, such as a click (e.g., you can't call `navigator.share` as part of the page load)
- you can also share any URL, not just URLs under your website's current scope: and you may also share `text` without a URL
- you should feature-detect it in case it's not available on your users' platform (e.g., via `navigator.share !== undefined`)

The URL

For the initial launch on Android, users using the Web Share API will be on a mobile device. Some sites might have a "m." URL, or a custom URL for the user's context. You can share any URL via the Web Share API, but you could reuse a canonical URL on your page to provide a better experience to the user. For example, you might do:

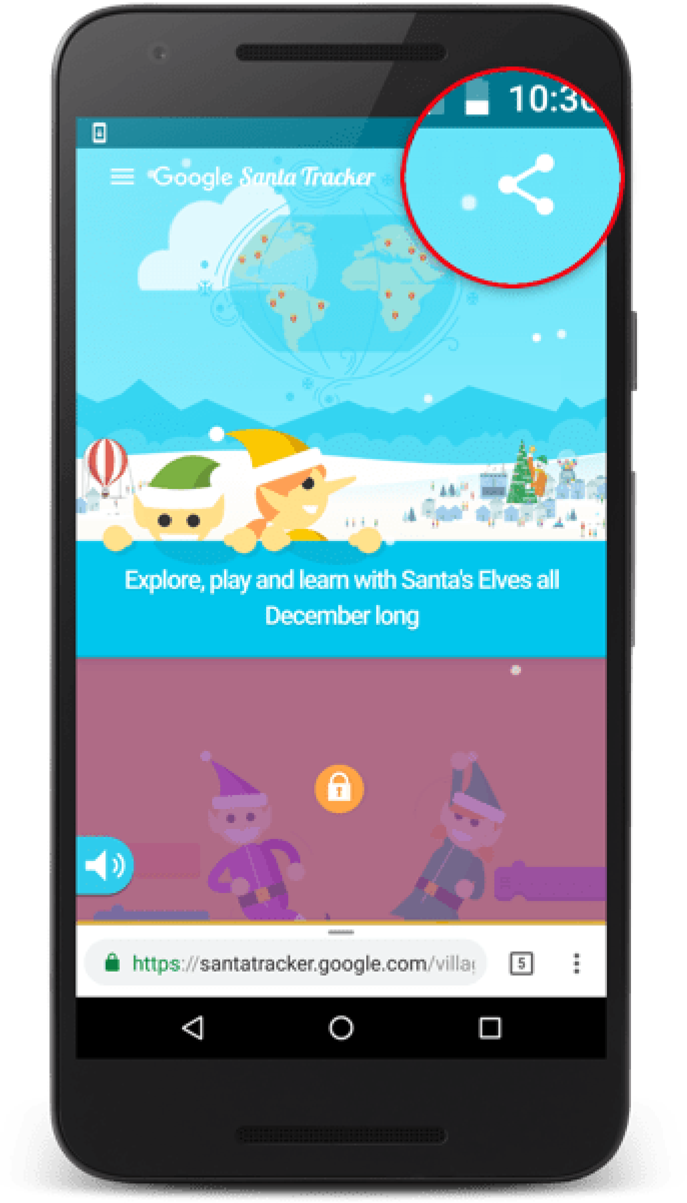
```
let url = document.location.href;
const canonicalElement = document.querySelector('link[rel=canonical]');
if (canonicalElement !== null) {
    url = canonicalElement.href;
}
navigator.share({url: url});
```

Case Study

Santa Tracker is a holiday tradition here at Google. Every December, you can celebrate the season with games and educational experiences: and in the new year, Santa Tracker is open-sourced and delivered.

In 2016, we used the Web Share API on Android via an Origin Trial (note: this is **not** required to use the Web Share API now, as part of Chrome 61). This API was a perfect fit for mobile—

in previous years, we had disabled share buttons on mobile, as space is at a premium and we couldn't justify having several share targets.


10:3
Google Santa Tracker
Explore, play and learn with Santa's Elves all December long
https://santatracker.google.com/villa
5

With the Web Share API, we were able to present just one button, saving precious pixels. We also found that users shared with Web Share around 20% more than users without the API enabled.

(If you're on Chrome 61 on Android, head to Santa Tracker and see Web Share in action.)

History

The Web Share API was originally launched as an Origin Trial as part of Chrome 55.

Prior to the Web Share API, there have been a number of ways to invoke native sharing capabilities on the platform, but they all have significant drawbacks. There was:

- Web Intents (dead)
- Protocol handling via `registerProtocolHandler`, but this has zero support on mobile
- Direct sharing to a well-known service URL such as Twitter
- Android intent: URL syntax (which was, unfortunately, Android-only, and required apps to opt-in)

More Information

Read more about the launch at Chrome Platform Status. Here are some important links:

- Sample
- Share explainer
- Intent to implement
- Discussion on Discourse

In the future, websites themselves will be allowed to register themselves as a "share receiver", enabling sharing *to* the web—from both the web and native apps. We are on the Chrome team are incredibly excited by this.

Last updated July 2, 2018.