

New in Chrome 68



By Pete LePage

Pete is a Developer Advocate

- The Add to Home Screen behavior on Android is changing to give you more control.
- The Page Lifecycle API tells you when your tab has been suspended or restored.
- And the Payment Handler API makes it possible for web-based payment apps to support the Payment Request experience.

And there's plenty more!

I'm Pete LePage. Let's dive in and see what's new for developers in Chrome 68!

Note: Want the full list of changes? Check out the [Chromium source repository change list](#).

Add to Home Screen changes

If your site meets the add to home screen criteria, Chrome will no longer show the add to home screen banner. Instead, you're in control over when and how to prompt the user.

To prompt the user, listen for the `beforeinstallprompt` event, then, save the event and add a button or other UI element to your app to indicate it can be installed.

```
let installPromptEvent;
```

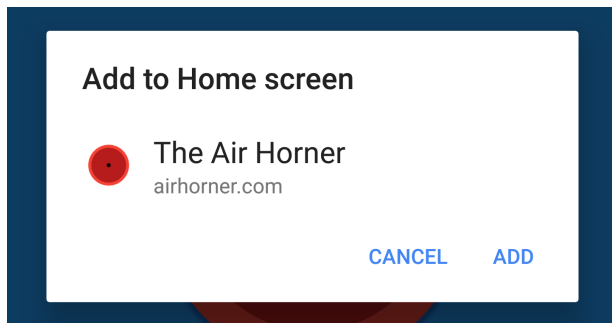
```
window.addEventListener('beforeinstallprompt', (event) => {  
  // Prevent Chrome <= 67 from automatically showing the prompt
```



```

event.preventDefault();
// Stash the event so it can be triggered later.
installPromptEvent = event;
// Update the install UI to notify the user app can be installed
document.querySelector('#install-button').disabled = false;
});

```

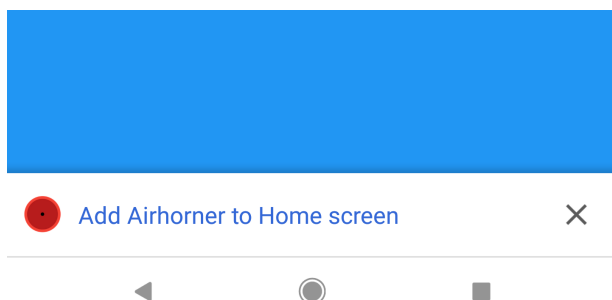


When the user clicks the install button, call `prompt()` on the saved `beforeinstallprompt` event, Chrome then shows the add to home screen dialog.

```

btnInstall.addEventListener('click', () => {
  // Update the install UI to remove the install button
  document.querySelector('#install-button').disabled = true;
  // Show the modal add to home screen dialog
  installPromptEvent.prompt();
  // Wait for the user to respond to the prompt
  installPromptEvent.userChoice.then(handleInstall);
});

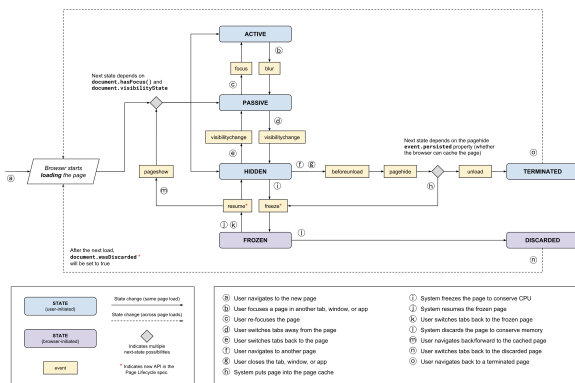
```



To give you time to update your site Chrome will show a mini-infobar the first time a user visits a site that meets the add to home screen criteria. Once dismissed, the mini-infobar will not be shown again for a while.

Changes to Add to Home Screen Behavior has the full details, including code samples you can use and more.

Page Lifecycle API



When a user has a large number of tabs running, critical resources such as memory, CPU, battery and the network can be oversubscribed, leading to a bad user experience.

If your site is running in the background, the system may suspend it to conserve resources. With the new Page Lifecycle API, you can now listen for, and respond to these events.

For example, if a user's had a tab in the background for a while, the browser may choose to suspend script execution on that page to conserve resources. Before doing so, it will fire the **freeze** event, allowing you to close open IndexedDB or network connections or save any unsaved view state. Then, when the user refocuses the tab, the **resume** event is fired, where you can reinitialize anything that was torn down.

```
const prepareForFreeze = () => {
  // Close any open IndexedDB connections.
  // Release any web locks.
  // Stop timers or polling.
};

const reInitializeApp = () => {
  // Restore IndexedDB connections.
  // Re-acquire any needed web locks.
  // Restart timers or polling.
};

document.addEventListener('freeze', prepareForFreeze);
document.addEventListener('resume', reInitializeApp);
```



Check out Phil's [Page Lifecycle API](#) post for **lots** more detail, including code samples, tips and more. You can find the [spec](#) and an [explainer doc](#) on GitHub.

Payment Handler API

The [Payment Request API](#) is an open, standards-based way to accept payments. The [Payment Handler API](#) extends the reach of Payment Request by enabling web-based payment apps to facilitate payments directly within the Payment Request experience.

As a seller, adding an existing web-based payment app is as easy as adding an entry to the `supportedMethods` property.

```
const request = new PaymentRequest([{\n  // Your custom payment method identifier comes here\n  supportedMethods: 'https://bobpay.xyz/pay'\n}], {\n  total: {\n    label: 'total',\n    amount: { value: '10', currency: 'USD' }\n  }\n});
```



If a service worker that can handle the specified payment method is installed, it will show up in the Payment Request UI and the user can pay with it.

Eiji has a [great post](#) that shows how to implement this for merchant sites, and for payment handlers.

And more!

These are just a few of the changes in Chrome 68 for developers, of course, there's plenty more.

- Content embedded in an `iframe` [requires a user gesture to navigate](#) the top-level browsing context to a different origin.
- Since Chrome 1, the CSS [cursor values for grab and grabbing](#) have been prefixed, we now support the standard, un-prefixed values. **Finally.**
- And - this is a big one! The HTTP [cache is now ignored when requesting updates to a service worker](#), bringing Chrome inline with the spec and other browsers.

New in DevTools

Be sure to check out [New in Chrome DevTools](#), to learn what's new in for DevTools in Chrome 68.

Subscribe

Then, click the [subscribe](#) button on 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 69 is released, I'll be right here to tell you – what's new in Chrome!

Feedback

Was this page helpful?

Great! Thank you for the feedback.

Sorry to hear that. Please [open an issue](#) and tell us how we can improve.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated July 24, 2018.