Changes to Add to Home Screen Behavior

By Pete LePage

Pete is a Developer Advocate

Since we first launched the add to home screen banner, we've been working to label Progressive Web Apps more clearly and simplify the way users can install them. Our eventual goal is to provide an install button in the omnibox across all platforms, and in Chrome 68 we are making changes towards that goal.
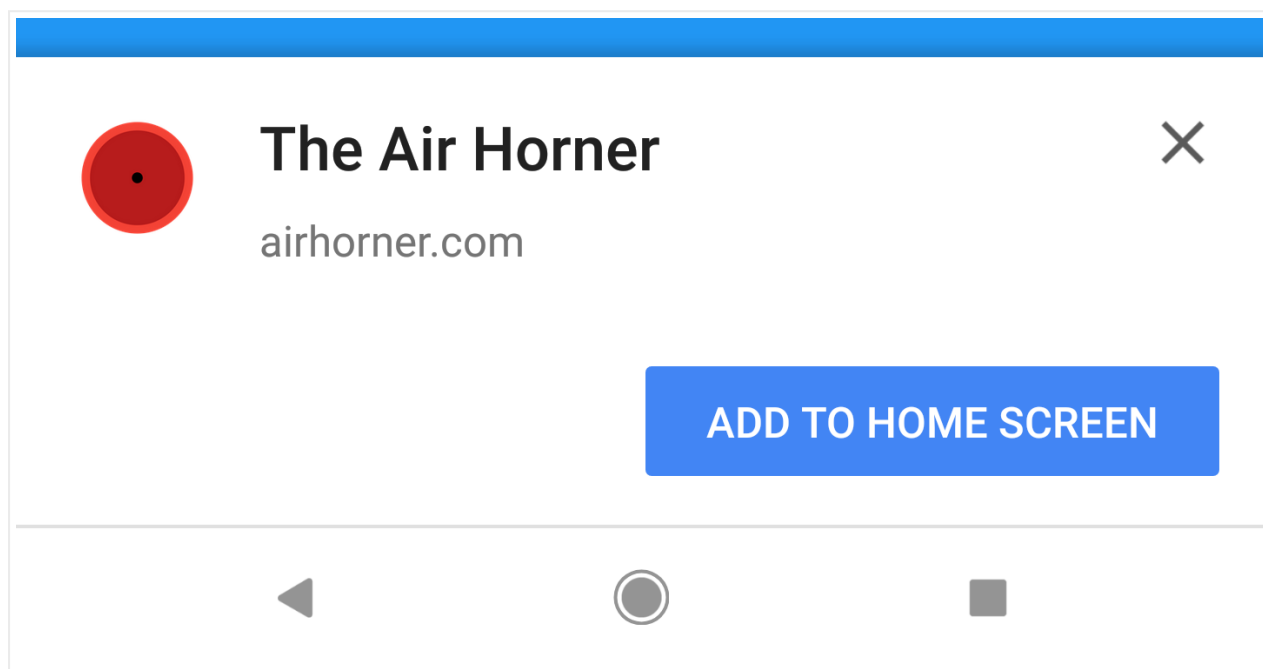
Note: Chrome 68 is out of beta. See the Add to Home Screen docs for the current behavior.

What's changing?

Starting in Chrome 68 on Android (Stable in July 2018), Chrome will no longer show the add to home screen banner. If the site meets the add to home screen criteria, Chrome will show the mini-infobar. Then, if the user clicks on the mini-infobar, or you call `prompt()` on the `beforeinstallprompt` event from within a user gesture, Chrome will show a modal add to home screen dialog.

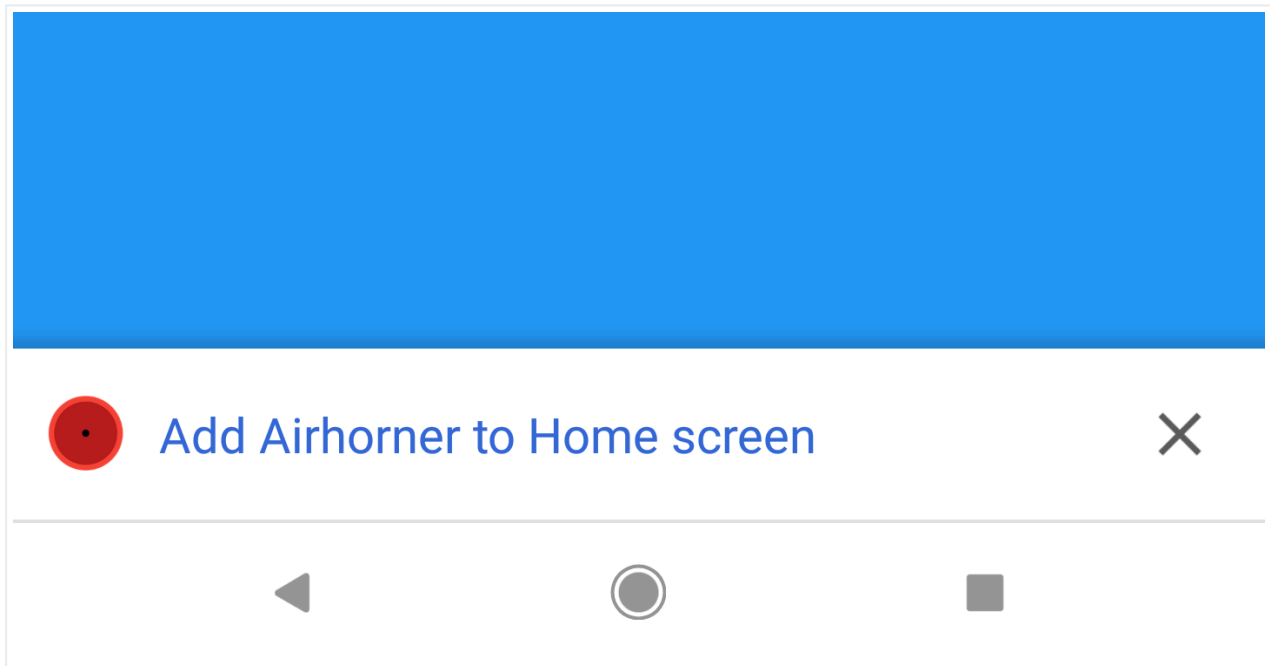**A2HS banner**

Chrome 67 and before

Shown automatically when site meets the add to home screen criteria, and the site does not call `preventDefault()` on the `beforeinstallprompt` event

OR

Shown by calling `prompt()` on the `beforeinstallprompt` event.

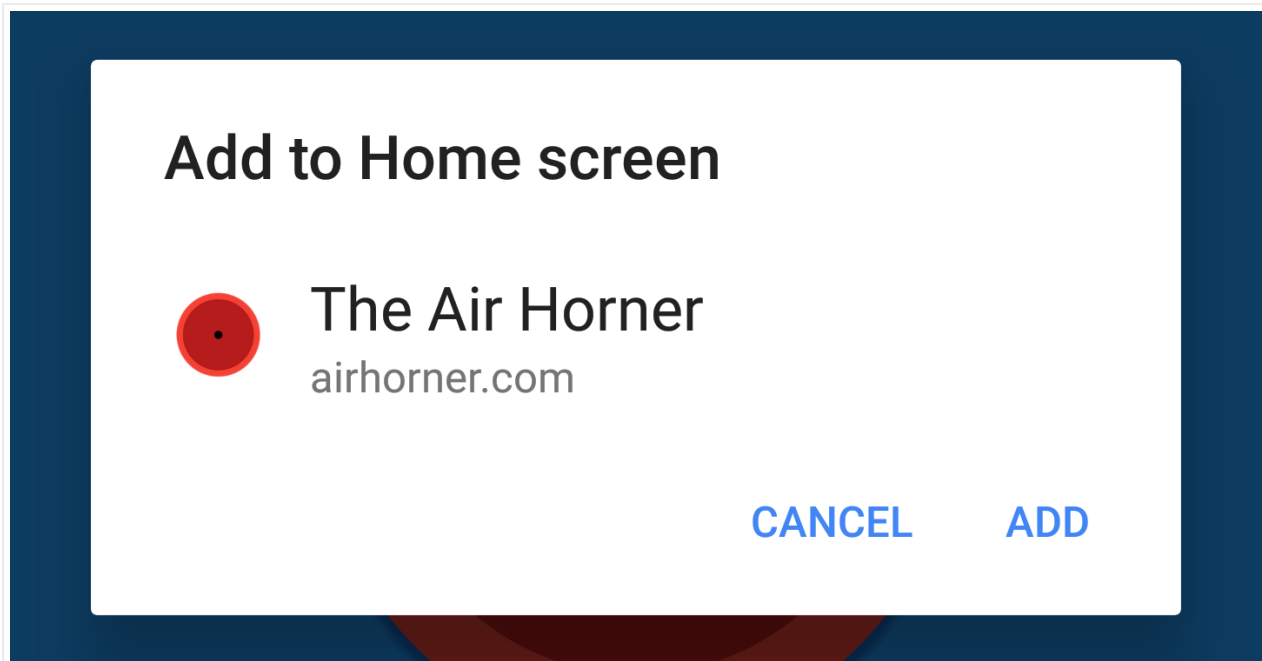**Mini-infobar**
Chrome 68 and later

Shown when the site meets the add to home screen criteria

If dismissed by a user, it will not be shown until a sufficient period of time (~3 months) has passed.

Shown regardless of if `preventDefault()` was called on the `beforeinstallprompt` event.

This UI treatment will be removed in a future version of Chrome when the omnibox install button is introduced.

A2HS Dialog

Shown by calling `prompt()` from within a user gesture on the `beforeinstallprompt` event in Chrome 68 and later.
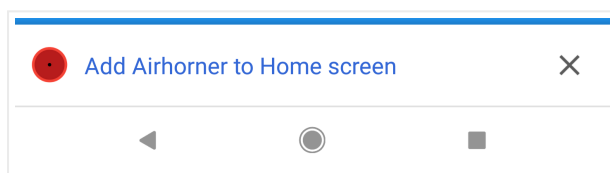
OR

Shown when a user taps the mini-infobar in Chrome 68 and later.
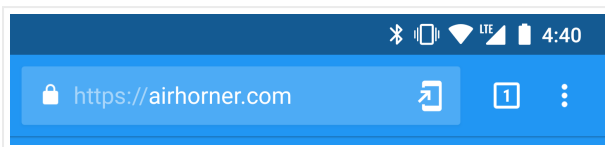
OR

Shown after the user clicks 'Add to Home screen' from the Chrome menu in all Chrome versions.

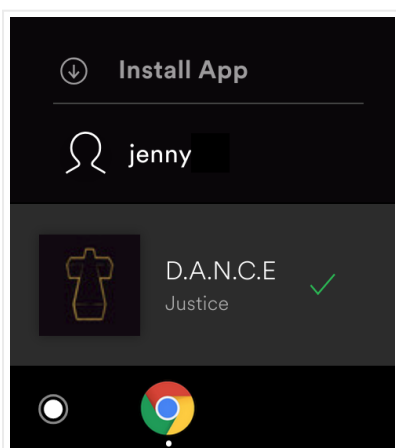## The mini-infobar

The mini-infobar

The mini-infobar is a Chrome UI component and is not controllable by the site, but can be easily dismissed by the user. Once dismissed by the user, it will not appear again until a sufficient amount of time has passed (currently 3 months). The mini-infobar will appear when the site meets the add to home screen criteria, regardless of whether you `preventDefault()` on the `beforeinstallprompt` event or not.

Early concept of the install button in the omnibox

The mini-infobar is an interim experience for Chrome on Android as we work towards creating a consistent experience across all platforms that includes an install button into the omnibox.

Triggering the add to home screen dialog

Install button on a Desktop Progressive Web App

Instead of prompting the user on page load (an anti-pattern for permission requests), you can indicate your app can be installed with some UI, which will then show the modal install prompt. For example this desktop PWA adds an 'Install App' button just above the user's profile name.

Prompting to install your app on a user gesture feels less spammy to the user and increases the likelihood that they'll click 'Add' instead of 'Cancel'. Incorporating an Install button into your app means that even if the user chooses not to install your app today, the button will still be there tomorrow, or whenever they're ready to install.

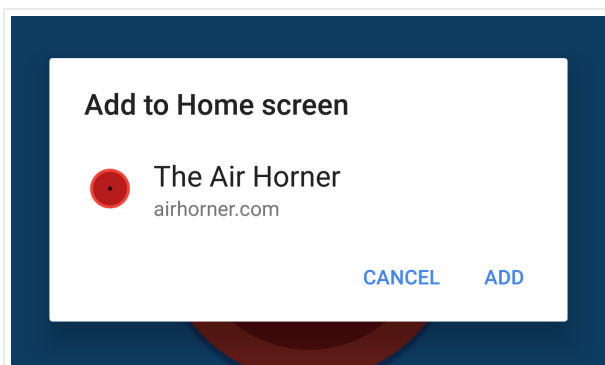## Listening for the `beforeinstallprompt` event

If your site meets the <u>add to home screen criteria</u>, Chrome will fire a `beforeinstallprompt` event, save a reference to the event, and update your user interface to indicate that the user can add your app to their home screen.

```
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;
});
```

Note: Your site must meet the add to home screen criteria in order for the `beforeinstallprompt` event to be fired and your app installed.

The `beforeinstallprompt` event will not be fired if the app is already installed (see the <u>add to home screen criteria</u>). But if the user later uninstalls the app, the `beforeinstallprompt` event will again be fired on each page navigation.

Showing the dialog with `prompt()`

Add to home screen dialog

To show the add to home screen dialog, call `prompt()` on the saved event from within a user gesture. Chrome will show the modal dialog, prompting the user to add your app to their home screen. Then, listen for the promise returned by the `userChoice` property of the `beforeinstallprompt` event. The promise returns an object with an `outcome` property after the prompt has shown and the user has responded to it.

```
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((choice) => {
    if (choice.outcome === 'accepted') {
      console.log('User accepted the A2HS prompt');
    } else {
      console.log('User dismissed the A2HS prompt');
    }
    // Clear the saved prompt since it can't be used again
    installPromptEvent = null;
  });
});
```

Note: Although the `beforeinstallprompt` event may be fired without a user gesture, calling `prompt()` requires one.

You can only call `prompt()` on the deferred event once, if the user clicks cancel on the dialog, you'll need to wait until the `beforeinstallprompt` event is fired on the next page navigation. Unlike traditional permission requests, clicking cancel will not block future calls to `prompt()` because it call must be called within a user gesture.

Additional Resources

Check out App Install Banners for more information, including:

- Details on the `beforeinstallprompt` event
- Tracking the user's response to the add home screen prompt
- Tracking if the app has been installed
- Determining if your app is running as an installed app

Last updated July 24, 2018.