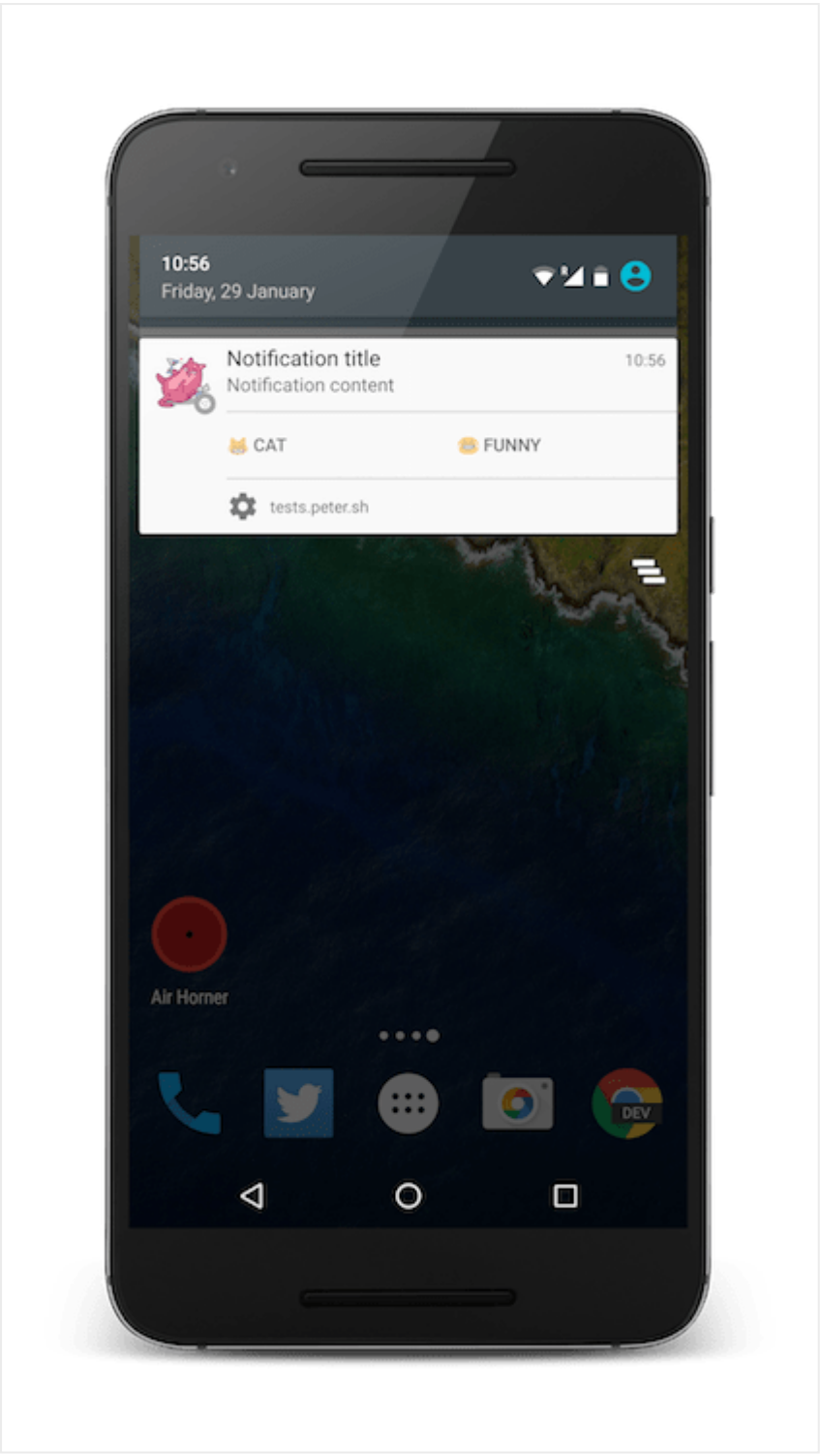
# Notification Actions in Chrome 48

**By** Paul Kinlan
Paul is a Developer Advocate

Early in 2015 we introduced Push Messaging and Notification in to Chrome for Android and Desktop. It was a great step forward on the web. Users could start to engage more deeply with experiences on the web even when the browser was closed.

Whilst it is great that you can send these messages, the only thing you could do with one was either to click it and open a page or dismiss it entirely.

If you look at the notifications provided natively to apps on mobile platforms such as iOS and Android, they each let the developer define contextual actions that the user can invoke and interact with. In Chrome 48 we have now added a similar ability to Web Notifications across Desktop and Chrome for Android.




The addition to the API is pretty simple. You just need to create an Array of actions and add them into the `NotificationOptions` object when you call `showNotification` from a ServiceWorker registration (either directly in the ServiceWorker or on a page via `navigator.serviceWorker.ready`).

Currently Chrome only supports two actions on each notification. Some platforms might be able to support more, and some platforms may support less or none at all. You can

determine what the platform supports by checking **`Notification.maxActions`**. In the following examples we are assuming the platform supports two actions.

```
self.registration.showNotification('New message from Alice', {
  actions: [
   {action: 'like', title: 'Like'},
   {action: 'reply', title: 'Reply'}]
});
```

This will create a simple notification with two buttons. Note, it is not possible to add icons to the action directly (yet), but you can use Emoji and the extended Unicode character set to add more context to your notifications buttons.

For example:

```
self.registration.showNotification("New message from Alice", {
  actions: [
   {action: 'like', title: '👍Like'},
   {action: 'reply', title: '⤻ Reply'}]
});
```

Now that you have created a notification and made it look 😈, the user may interact with the notification at some time in the future. Interactions with the notification all currently (as of Chrome 48) come through the **`notificationclick`** event registered in your service worker and they can either be a general click on the notification or a specific tap on one of the action buttons. Side note, in the future you will also be able to respond to the notificationclose event.

To understand which action the user took you need to inspect the **`action`** property on the event and then you have the choice of either opening a new page for the user to complete an action or to perform the task in the background.

```
self.addEventListener('notificationclick', function(event) {
  var messageId = event.notification.data;

  event.notification.close();

  if (event.action === 'like') {
    silentlyLikeItem();
  }
  else if (event.action === 'reply') {
    clients.openWindow("/messages?reply=" + messageId);
  }
  else {
    clients.openWindow("/messages?reply=" + messageId);
```

```
  }
}, false);
```

The interesting thing is that the actions don't have to open up a new window, they can perform general application interactions without creating a user interface. For example, a user could "Like" or "Delete" a social media post that would perform the action on the user's local data and then synchronize it with the cloud without opening a UI (although it is good practice to message the data change to any open windows so the UI can be updated). For an action that requires user interaction you would open a window for the user to reply.

Because platforms will not support the same number of actions, or in some cases not be able to support Notification Action buttons at all, you will need to ensure that you always provide a sensible fallback to a task that is what you would expect the user to do if they were to just click the notification.

If you want to see this in action today, check out Peter Beverloo's Notification Test Harness and read up on the Notifications specification or follow along with spec as it updates.

**Note:** Be sure to check out the full documentation including best practices for using Web Push Notifications

---



*Last updated July 2, 2018.*