

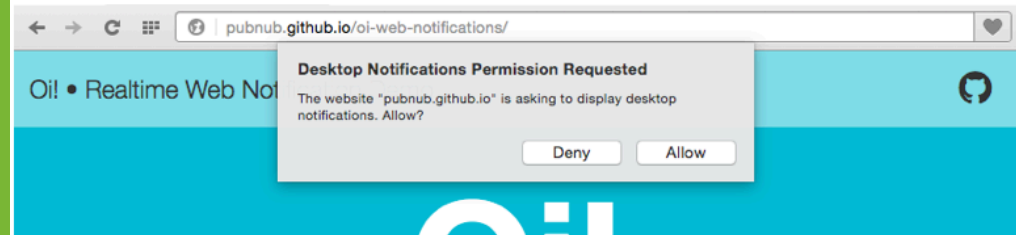
Notifications

*Keep your user informed
with Web Notifications*

John Feiner

Requesting permission

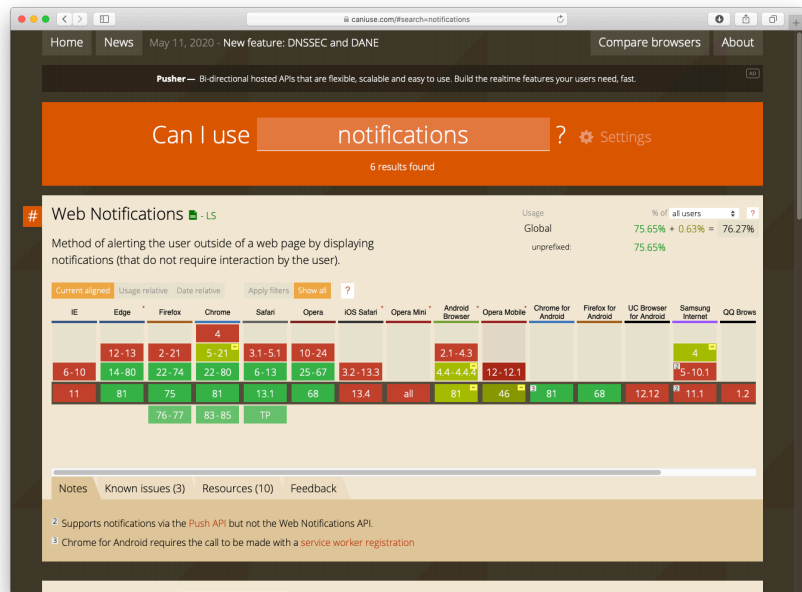
Each browser provides a built-in permission UI for the Web Notification API, so that you can control which web pages can send you notifications. When your page is loaded for the first time, a browser-specific permission dialog is shown to the user. After the user has granted permission, your page can send notifications, otherwise (denied), it can't.



```
Notification.requestPermission(function() {  
  if (Notification.permission === 'granted') {  
    // Now your page can send the user notifications!  
  }  
});
```

<https://dev.opera.com/articles/web-notifications-pubnub/>

Available?



Web APIs

[Web technology for developers](#) > [Web APIs](#)

English ▼

When writing code for the Web, there are a large number of Web APIs available. Below is a list of all the APIs and interfaces (object types) that you may be able to use while developing your Web app or site.

Web APIs are typically used with JavaScript, although this doesn't always have to be the case.

Specifications

This is a list of all the APIs that are available.

A
Ambient Light Events

B
Background Tasks
Battery API
Beacon
Bluetooth API
Broadcast Channel API

C
CSS Counter Styles
CSS Font Loading API
CSSOM
Canvas API
Channel Messaging API
Console API
Credential Management API

D
DOM

E
Encoding API
Encrypted Media Extensions

F
Fetch API
File System API
Frame Timing API
Fullscreen API

G
Gamepad API
Geolocation API

H
HTML Drag and Drop API
High Resolution Time
History API

I
Image Capture API
IndexedDB
Intersection Observer API

L
Long Tasks API

M
Media Capabilities API
Media Capture and Streams

Media Session API
Media Source Extensions
MediaStream Recording

N
Navigation Timing
Network Information API

P
Page Visibility API
Payment Request API
Performance API
Performance Timeline API
Permissions API
Pointer Events
Pointer Lock API
Proximity Events
Push API

R
Resize Observer API
Resource Timing API

S
Server Sent Events
Service Workers API
Storage

Storage Access API
Streams

T
Touch Events

U
URL API

V
Vibration API

W
Web Animations
Web Audio API
Web Authentication API
Web Crypto API
Web Notifications
Web Storage API
Web Workers API
WebGL
WebRTC
WebVR API
WebVTT
WebXR Device API
Websockets API

§ 3.5.2. Using actions from a service worker

Persistent notifications fire `notificationclick` events on the `ServiceWorkerGlobalScope`.

Here a service worker shows a notification with a single "Archive" action, allowing users to perform this common task from the notification without having to open the website (for example the notification platform might show a button on the notification). The user can also activate the main body of the notification to open their inbox.

Example

```
self.registration.showNotification("New mail from Alice", {
  actions: [{action: 'archive', title: "Archive"}]
});

self.addEventListener('notificationclick', function(event) {
  event.notification.close();
  if (event.action === 'archive') {
    silentlyArchiveEmail();
  } else {
    clients.openWindow("/inbox");
  }
}, false);
```

Tutorial

Service worker

Ask for permissions

Display message

A little notification anatomy

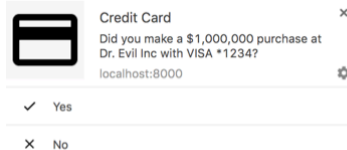
In the next section we're going to throw a bunch of pictures at you, but we promised code. So, here it is. With a service worker registration you call `showNotification` on a registration object.

```
serviceWorkerRegistration.showNotification(title, options);
```

The `title` argument appears as a heading in the notification. The `options` argument is an object literal that sets the other properties of a notification. A typical options object looks something like this:

```
{
  "body": "Did you make a $1,000,000 purchase at Dr. Evil...",
  "icon": "images/ccard.png",
  "vibrate": [200, 100, 200, 100, 200, 100, 400],
  "tag": "request",
  "actions": [
    { "action": "yes", "title": "Yes", "icon": "images/yes.png" },
    { "action": "no", "title": "No", "icon": "images/no.png" }
  ]
}
```

This code produces a notification like the one in the image. It generally provides the same capabilities as a native application. Before diving into the specifics of implementing those capabilities, I'll show you how to use those capabilities effectively. We'll go on to describe the mechanics of implementing push



<https://developers.google.com/web/fundamentals/codelabs/push-notifications>

<https://developers.google.com/web/fundamentals/push-notifications/>