

Setup	Subscribe the user (cont)	Notification click
Clone repo <code>git clone https://github.com/GoogleChrome/push-notifications.git</code>	<code>});</code>	<code>self.addEventListener('notification click</code>
Install webserver <code>https://chrome.google.com/webstore/detail/web-server-for-chrome/ofhbbkphhbklhfoeikjpcbhmlcgigb</code>	Get Application Server Keys	<code>function(event) {</code> <code> console.log('[Service Worker</code> <code> notification click Received.');</code> <code> event.notification.close();</code> <code> event.waitUntil(</code>
Select app folder from cloned repo in webserver	Generate public/private keys <code>https://web-push-codelab.glitch.me/</code>	<code>clients.openWindow('https://rs.google.com/web/')</code>
Always update the service worker Open DevTools (Right Click > Inspect) -> Application -> Service Workers -> Update on Reload	Insert them into your application	<code>);</code> <code>});</code>
Register a Service Worker <code>navigator.serviceWorker.register('sw.js')</code> <code>.then(function(swReg) { console.log('Service Worker is registered', swReg);</code> <code> swRegistration = swReg; })</code>	Handle Permission Denied <code>if (Notification.permission === 'denied') {</code> <code> pushButton.textContent = 'Push Messaging Blocked.';</code> <code> pushButton.disabled = true;</code> <code> updateSubscriptionOnServer(null);</code> <code> return;</code> <code>}</code>	Sending push messages Use <code>https://web-push-codelab.glitch.me/message</code>
Subscribe the user <code>swRegistration.pushManager.getSubscription().then(function(subscription) {</code> <code> isSubscribed = !(subscription === null);</code> <code> updateSubscriptionOnServer(subscription);</code> <code> const applicationServerKey = urlB64ToUint8Array(applicationServerPublicKey);</code> <code> swRegistration.pushManager.subscribe({</code> <code> userVisibleOnly: true,</code> <code> applicationServerKey: applicationServerKey</code> <code> })</code> <code>.then(function(subscription) {</code> <code> updateSubscriptionOnServer(subscription);</code> <code> isSubscribed = true;</code> <code> })</code>	<code> if (isSubscribed) {</code> <code> pushButton.textContent = 'Disable Push Messaging';</code> <code> } else {</code> <code> pushButton.textContent = 'Enable Push Messaging';</code> <code> }</code> <code> pushButton.disabled = false;</code>	Unsubscribe the user <code>pushButton.addEventListener('click', function() {</code> <code> pushButton.disabled = true;</code> <code> if (isSubscribed) {</code> <code> unsubscribeUser();</code> <code> } else {</code> <code> subscribeUser();</code> <code> }</code> <code>});</code> <code>swRegistration.pushManager.getSubscription().then(function(subscription) {</code> <code> if (subscription) {</code> <code> return subscription.unsubscribe();</code> <code> }</code> <code> })</code> <code>.then(function() {</code> <code> updateSubscriptionOnServer();</code> <code> console.log('User is unsubscribed = false');</code> <code> updateBtn();</code> <code> })</code>
	Handle a Push Event	

```
self.addEventListener('push',
function(event) {
  console.log('[Service
Worker] Push Received. ');
  console.log([Service
Worker] Push had this data:
"${event.data.text()}");
  const title = 'Push
Codelab';
  const options = {
    body: 'Yay it works.',
    icon: 'images/icon.png',
    badge: 'images/badg-
e.png'
  };
  event.waitUntil(self.reg-
istration.showNotification(-
title, options));
});
```



By **[deleted]**

cheatography.com/deleted-96251/

Published 21st January, 2020.

Last updated 22nd January, 2020.

Page 1 of 2.

Sponsored by **Readable.com**

Measure your website readability

<https://readable.com>