

Aim : To code and register a service worker, and complete the install and activation process for a new service worker for the E-commerce PWA.

Theory :

Service Worker

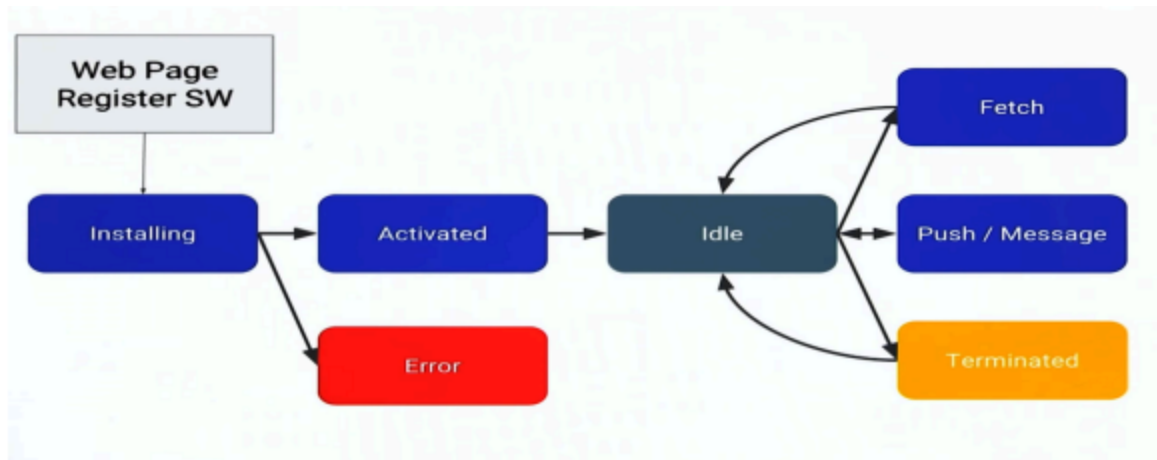
Service Worker is a powerful script that operates in the background of a browser, independently of user interaction. It acts as a network proxy, intercepting outgoing HTTP requests made by your web application. With Service Worker, you can manage network traffic, handle push notifications, and develop "offline first" web applications using the Cache API.

- **Network Proxy:** Service workers intercept all outgoing HTTP requests, allowing you to handle these requests. For example, you can serve content from a local cache if available, improving performance and providing a better user experience.
- **Offline Capabilities:** Service workers enable offline functionality by caching essential application resources such as HTML, CSS, JavaScript, and images. When a user is offline, the service worker can retrieve the requested content from the cache, ensuring a seamless experience even without an internet connection.
- **HTTPS Requirement:** For security reasons, service workers require HTTPS connections. This ensures secure communication between the service worker, your application, and the server.

What can we do with Service Workers?

- **Network Traffic Control:** Manage all network traffic of the page and manipulate requests and responses. For example, you can respond to a CSS file request with plain text or an HTML file request with a PNG file. You can also respond with the requested content.
- **Caching:** Cache any request/response pair with Service Worker and Cache API, allowing you to access offline content anytime.
- **Push Notifications:** Manage push notifications with Service Worker, enabling you to show informational messages to the user.
- **Background Processes:** Even when the internet connection is broken, you can start any process with the Background Sync feature of Service Worker.

Service Worker Cycle



Steps for coding and registering a service worker for your E-commerce PWA completing the install and activation process:

1. Create the Service Worker File (sw.js):
2. Register the Service Worker:

In your main JavaScript file (e.g., main.js or app.js), add the following code:



```
> index.html {} manifest.json JS serviceworker.js X
JS serviceworker.js > self.addEventListener("install") callback
1  var staticCacheName = "pwa-v1"; // Update cache name with versioning
2  self.addEventListener("install", function (e) {
3    e.waitUntil(
4      caches.open(staticCacheName).then(function (cache) {
5        return cache.addAll([
6          '/', // Add your homepage and other important routes here
7          '/index.html',
8          './assets/images/brand-1.png',
9          './assets/images/brand-2.png',
10         ]);
11       });
12     );
13   });
14   self.addEventListener('activate', function(event) {
15     event.waitUntil(
16       caches.keys().then(function(cacheNames) {
17         return Promise.all(
18           cacheNames.filter(function(cacheName) {
19             // Return true if you want to remove this cache, false otherwise
20             return cacheName.startsWith('pwa-') && cacheName !== staticCacheName;
21           }).map(function(cacheName) {
22             return caches.delete(cacheName);
23           });
24         );
25       });
26     );
27   });
28   self.addEventListener("fetch", function (event) {
29     event.respondWith(
30       caches.match(event.request).then(function (response) {
31         return response || fetch(event.request);
32       })
```

Now we need to connect the serviceworker.js to index.html for which we enter the following code into our index.html file in the script tag.

```

<script>
window.addEventListener('load', () => {
  registerSW();
});
async function registerSW() {
  if ('serviceWorker' in navigator) {
    try {
      await navigator.serviceWorker.register('/serviceworker.js', { scope: '/' });
      console.log('SW registration successful');
    } catch (error) {
      console.log('SW registration failed:', error);
    }
  } else {
    console.log('Service Worker is not supported in this browser.');
```

Output

The screenshot shows a web browser window with the URL 127.0.0.1:5500/index.html. The page displays a navigation menu with links like Home, Shop, Collections, Blogs, and Contact. Below the menu, there's a section for 'High Quality Pet Food' with a 'Sale up to 40% off today' and a 'Shop Now' button. The 'Top categories' section lists 'Cat Food', 'Cat Toys', 'Dog Food', and 'Dog Toys'. The Network tab is open, showing a list of requests. The requests include index.html, style.css, script.js, and various images. The status of the requests is shown as 'Finished' or 'Failed'.

Name	Status	Type	Initiator	Size	Time	Waterfall
ws	Finished	websocket	index.html:1067	0 B	Pending	
index.html	200	document	Other	(ServiceW...	3 ms	
style.css	(failed) ne...	stylesheet	index.html:23	0 B	5 ms	
css2?family=Bangers&family=Carter+O...	200	stylesheet	index.html:32	(memory c...	0 ms	
hero-banner.jpg	(failed) ne...		index.html:37	0 B	5 ms	
script.js	(failed) ne...	script	index.html:1036	0 B	8 ms	
ionicons.esm.js	200	script	index.html:1041	(ServiceW...	5 ms	
hero-banner.jpg	(failed) ne...	fetch	serviceworker.js:29	0 B	4 ms	
ionicons.esm.js	200	fetch	serviceworker.js:29	(disk cache)	1 ms	
script.js	(failed) ne...	fetch	serviceworker.js:29	0 B	1 ms	
p-e26ac56f.js	200	script	ionicons.esm.js:1	(ServiceW...	1 ms	
p-e26ac56f.js	200	fetch	serviceworker.js:29	(disk cache)	0 ms	
p-5c60b45e.entry.js	200	script	p-e26ac56f.js:1	(ServiceW...	1 ms	
p-5c60b45e.entry.js	200	fetch	serviceworker.js:29	(disk cache)	0 ms	
FeVQS08Tqb0h60ACH55Q2A.woff2	200	font	css2	(memory c...	0 ms	
q5uCsoe5IOB2-pXv9UcN8hA.woff2	200	font	css2	(memory c...	0 ms	
pe0TMImsLY8lv1o4X1M8ce2Cx3yop4t...	200	font	css2	(memory c...	0 ms	

Summary: 138 requests, 0 B transferred, 142 kB resources, Finish: 11.99 s, DOMContentLoaded: 17 ms, Load: 20 ms

Conclusion : I have understood and successfully registered a service worker, and completed the install and activation process for a new service worker for the E-commerce PWA.