

Experiment 8

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:

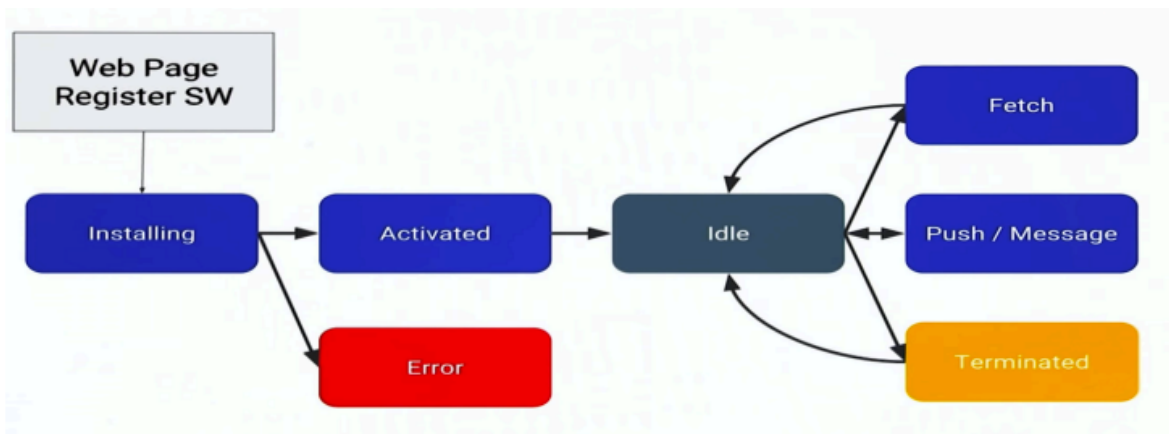
Service Worker:

A service worker is a type of web worker, a JavaScript file that runs in the background separate from the main browser thread. It enables functionalities such as push notifications, background sync, and caching, making Progressive Web Apps (PWAs) more reliable and engaging.

Key features of service workers include:

1. **Background Processing:** Service workers run independently of the main browser thread, allowing them to handle tasks such as fetching resources, intercepting network requests, and managing caches.
2. **Offline Capabilities:** By intercepting network requests, service workers can implement offline caching strategies, enabling PWAs to work even when the user is offline or has a poor network connection.
3. **Push Notifications:** Service workers can receive push notifications from a server, allowing PWAs to send timely updates or notifications to users, even when the PWA is not actively open in the browser.
4. **Improved Performance:** By caching resources and assets, service workers can improve the performance of PWAs by reducing network requests and enabling faster load times.

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 (service-worker.js)

```
JS service-worker.js > ...
1  // Service worker code
2
3  // Listen for the install event
4  self.addEventListener("install", function (event) {
5    console.log("Service Worker installed");
6  });
7
8  // Listen for the activate event
9  self.addEventListener("activate", function (event) {
10   console.log("Service Worker activated");
11 });
12
13 // Fetch event listener
14 self.addEventListener("fetch", function (event) {
15   // You can add caching strategies or other fetch handling here
16   console.log("Fetching:", event.request.url);
17 });
18
```

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

```
<script>
  if ("serviceWorker" in navigator) {
    window.addEventListener("load", function () {
      navigator.serviceWorker.register("/service-worker.js").then(
        function (registration) {
          console.log(
            "Service Worker registered with scope:",
            registration.scope
          );
        },
        function (err) {
          console.log("Service Worker registration failed:", err);
        }
      );
    });
  }
</script>
```

Output

The screenshot displays the Chrome DevTools interface with the 'Service workers' panel selected in the left sidebar. The main panel shows details for a service worker registered at `http://127.0.0.1:5500/`. The source is `service-worker.js`, and it was received on 3/25/2024 at 6:19:44 PM. The status is '#3375 activated and is running', with a 'stop' button. Below this, there are input fields for 'Push' (containing 'Test push message from DevTools.') and 'Sync' (containing 'test-tag-from-devtools'), each with a corresponding button. A 'Periodic Sync' section also has an input field with 'test-tag-from-devtools' and a 'Periodic Sync' button. An 'Update Cycle' table shows the worker's lifecycle: Version #3375, Install (completed), Wait (in progress), and Activate (pending). The bottom of the panel shows another instance of the service worker, which is '#3373 trying to install' and was received on 1/1/1970 at 5:30:00 AM.

Application

- Manifest
- Service workers**
- Storage

Storage

- Local storage
- Session storage
- IndexedDB
- Web SQL
- Cookies
- Private state tokens
- Interest groups
- Shared storage
- Cache storage

Background services

- Back/forward cache
- Background fetch
- Background sync
- Bounce tracking mitigation
- Notifications
- Payment handler
- Periodic background sync
- Speculative loads
- Push messaging
- Reporting API

Service workers

☐ Offline ☐ Update on reload ☐ Bypass for network

http://127.0.0.1:5500/ [Network requests](#) [Update](#) [Unregister](#)

Source [service-worker.js](#)

Received 3/25/2024, 6:19:44 PM

Status ● #3375 activated and is running [stop](#)

Push [Push](#)

Sync [Sync](#)

Periodic Sync [Periodic Sync](#)

Update Cycle

Version	Update Activity	Timeline
▶ #3375	Install	<div style="width: 100%;"></div>
▶ #3375	Wait	<div style="width: 20%;"></div>
▶ #3375	Activate	

http://127.0.0.1:5500/ [Network requests](#) [Update](#) [Unregister](#)

Source

Status ☐ #3373 trying to install Received 1/1/1970, 5:30:00 AM

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.