

Experiment N0-8

Code:

```
// Change this to a new version to trigger an update
const CACHE_NAME = "pwa-cache-v2";

const ASSETS_TO_CACHE = [
  "./",
  "./index.html",
  "./styles.css",
  "./app.js",
  "./manifest.json",
  "./images/shop_icon.png",
];

// Install event - Cache important assets
self.addEventListener("install", (event) => {
  console.log("✅ Service Worker Installed");

  // This line forces the waiting service worker to become active
  self.skipWaiting();

  event.waitUntil(
    caches.open(CACHE_NAME).then((cache) => {
      console.log("✅ Caching assets");
      return cache.addAll(ASSETS_TO_CACHE);
    })
  );
});

// Activate event - Cleanup old caches
self.addEventListener("activate", (event) => {
  console.log("✅ Service Worker Activated");

  // This ensures the service worker takes control immediately
  event.waitUntil(clients.claim());

  event.waitUntil(
    caches.keys().then((cacheNames) =>
```

```

    Promise.all(
      cacheNames.map((cache) => {
        if (cache !== CACHE_NAME) {
          console.log("🗑️ Deleting old cache:", cache);
          return caches.delete(cache);
        }
      })
    )
  );
});

```

// Fetch event - Modified to always check network first for HTML files

```

self.addEventListener("fetch", (event) => {
  const url = new URL(event.request.url);

  // Always get fresh HTML from network
  if (
    event.request.mode === "navigate" ||
    (event.request.method === "GET" &&
      event.request.headers.get("accept").includes("text/html"))
  ) {
    console.log("🔄 Getting fresh HTML for:", url.pathname);

    event.respondWith(
      fetch(event.request).catch(() => caches.match("./index.html"))
    );
  } else {
    // For non-HTML resources, use cache-first strategy
    event.respondWith(
      caches
        .match(event.request)
        .then((response) => {
          return (
            response ||
            fetch(event.request).then((fetchResponse) => {
              // Save new resources in cache
              const responseClone = fetchResponse.clone();
              caches
                .open(CACHE_NAME)
                .then((cache) => cache.put(event.request, responseClone));
            })
          );
        })
    );
  }
});

```

```

        return fetchResponse;
      })
    );
  })
  .catch(() => {
    // Fallback for offline
    if (event.request.url.includes(".html")) {
      return caches.match("./index.html");
    }
  })
);
}
});

// Rest of your code remains the same
self.addEventListener("sync", (event) => {
  if (event.tag === "sync-cart") {
    event.waitUntil(syncCart());
  }
});

self.addEventListener("push", (event) => {
  const options = {
    body: "This is a push notification!",
    icon: "/images/shop_icon.png",
    badge: "/images/badge.png",
  };
  event.waitUntil(self.registration.showNotification("New Message!", options));
});

async function syncCart() {
  // Logic to sync cart data with the server
  console.log("🔄 Syncing cart data with the server");
  // Example: Fetch unsynced cart data from IndexedDB and send to server
}

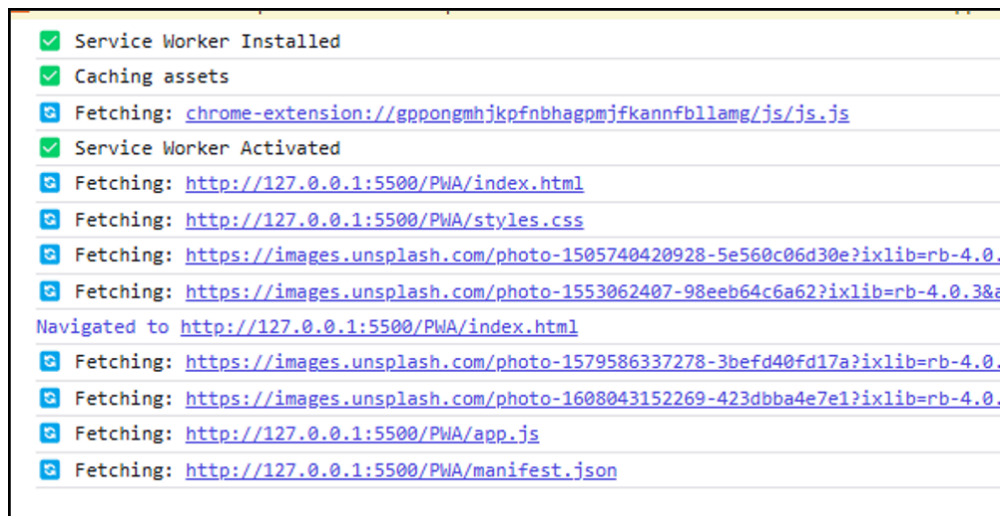
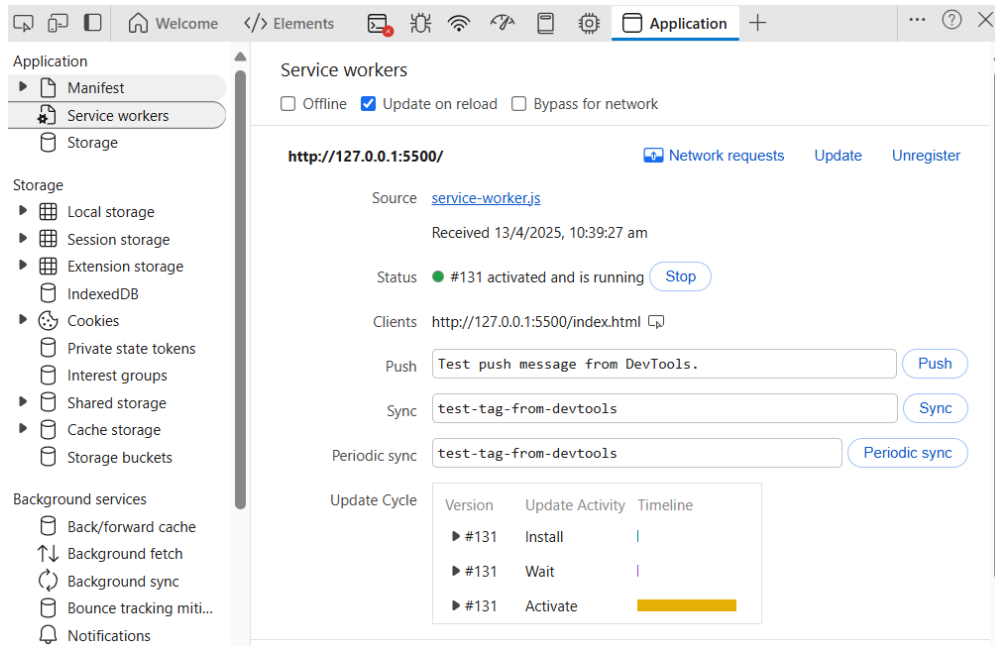
// Push event - Handle push notifications
self.addEventListener("push", (event) => {
  const data = event.data
    ? event.data.json()
    : { title: "New Notification", body: "Check out our latest products!" };

```

```

console.log("🔔 Push notification received:", data);
const options = {
  body: data.body,
  icon: "/PWA/icons/icon-192x192.png",
  badge: "/PWA/icons/icon-192x192.png",
};
event.waitUntil(self.registration.showNotification(data.title, options));
});

```



InspectorConsoleDebuggerNetworkStyle EditorPerformanceMemoryStorageAccessibility

database

St...M...DomainFileInitiatorTy...Transferr...Size

200GET127.0...database.jsonindex.ht...jsonservice w...2...

Filter HeadersBlockResend

GET http://127.0.0.1:5500/database.json

Status200 OK

VersionHTTP/1.1

Transferred0 B (2.24 kB size)

Referrer Policystrict-origin-when-cross-origin

DNS ResolutionSystem

Response Headers (351 B)Raw

Accept-Ranges: bytes

Access-Control-Allow-Credentials: true

Cache-Control: public, max-age=0

Connection: keep-alive

Content-Length: 2243

Content-Type: application/json; charset=UTF-8

Date: Sun, 13 Apr 2025 07:10:33 GMT

ETag: W/"8c3-1962d9e25aa"

Keep-Alive: timeout=5

Last-Modified: Sun, 13 Apr 2025 05:27:41 GMT

Vary: Origin

Request Headers (372 B)Raw

Accept: */*

Accept-Encoding: gzip, deflate, br, zstd

Accept-Language: en-US,en;q=0.5

Connection: keep-alive

Host: 127.0.0.1:5500

Referer: http://127.0.0.1:5500/index.html

1 request2.24 kB / 0 B transferredFinish: 248 msDOMContentLoaded: 41 m