

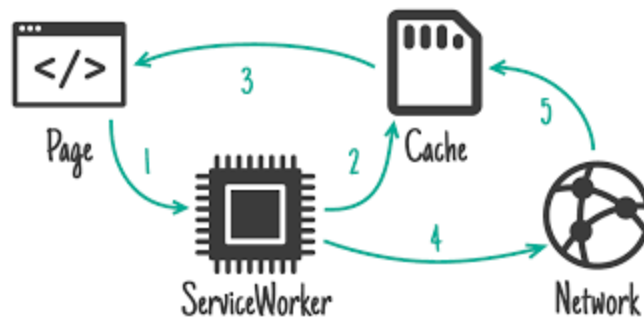
# Service Worker Caching

## ▼ What is a service worker?

Service workers act as proxy servers between web applications, the browser, and the network (when available). They are intended, among other things, to enable the creation of effective offline experiences, intercept network requests, and take appropriate action based on whether the network is available, and update assets residing on the server. (source MDN)

It is a script that runs in the background, separate from the web page which has abilities to intercept and handle network calls, caching and other offline functionalities. It acts like a proxy.

## ▼ How does service worker caching work?



1. Service worker intercepts request made by the page
2. Service worker checks the cache for the required requested resource
3. If the resource is available, it is returned to page from cache without making a network call
4. If the resource is unavailable, a network call is made and fetches it
5. This resource is set in cache and returned to the page

## ▼ How does sw.js file work?

We name a cache and all the URLs we want to cache. We need to add the URLs to the cache. We install the service worker, open a cache with our unique cache name and then add all the URLs one at a time.

We also added an event listener to the fetch call to listen to every network call made. We first check if the cache has the resources for the corresponding request. If we have it, we respond to the fetch call with the cache resource, else we continue with the fetch call and respond with the fetched resources

```
const CACHE_NAME = 'my-cache-v1'; // cache name to identify it
const urlsToCache = [
  '/',
  '/index.html',
  '/styles.css',
  '/app.js',
  '/image.gif'
]; // these are the URLs whose request we want to cache

self.addEventListener('install', event => {
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then(cache => cache.addAll(urlsToCache))
  );
}); // sw does not finish installing until the cache is created

self.addEventListener('fetch', event => { // listen to every
  event.respondWith(
    caches.match(event.request) // check if we have the resource
      .then(response => {
        return response || fetch(event.request); // if not, fetch it
      })
  );
});
```

### ▼ How do we register the service worker?

We need to check whether service worker is supported in the browser. if service worker is supported, we register the service worker file `sw.js`. The service worker file opens a cache named `'my-cache-v1'` and caches all the urls given( `/`, `/index.html`, `/styles.css`, `/app.js`, `/image.gif` ). For every network request made, it checks the cache for the requested resources. If found, we respond with the caches resource, otherwise we fetch the request, cache it, and respond with the fetched data.

```
<script>
  if ("serviceWorker" in navigator) {
    navigator.serviceWorker
      .register("/sw.js")
      .then((registration) => {
        console.log(
          "Service Worker registered with scope:",
          registration.scope
        );
      })
      .catch((error) => {
        console.error("Service Worker registration failed");
      });
  }
</script>
```