

PROGRESSIVE WEB APPS

---

**SERVICE WORKERS**

## WHAT IS A SERVICE WORKER

- ▶ A service worker is a script that your browser runs in the background, separate from a web page

## SERVICE WORKERS

- ▶ Run separate from a HTML page
- ▶ Lives on even pages have been closed
- ▶ Can access cache storage, network request, web notification
- ▶ Listen to specific events ex. fetch, install, activate

# SERVICE WORKERS

- ▶ Run separate from a HTML page
- ▶ Lives on even pages have been closed
- ▶ Can access cache storage, network request, web notification
- ▶ Listen to specific events ex. fetch, install, activate

# SERVICE WORKERS

- ▶ Run separate from a HTML page
- ▶ Lives on even pages have been closed
- ▶ Can access cache storage, network request, web notification
- ▶ Listen to specific events ex. fetch, install, activate

## SERVICE WORKERS

- ▶ Run separate from a HTML page
- ▶ Lives on even pages have been closed
- ▶ Can access cache storage, network request, web notification
- ▶ Listen to specific events ex. fetch, install, activate

### COMMON USED

- ▶ Push notifications
- ▶ Background sync
- ▶ Offline mode

### NOTE

- ▶ Service workers can't access DOM



# PREREQUISITES

- ▶ Browser support
  - ▶ Chrome, Firefox - Shipped
  - ▶ Safari, Edge - In development
  - ▶ <https://jakearchibald.github.io/isserviceworkerready>
- ▶ HTTPS setup on server.
- ▶ HTTPS host - [Firebase hosting](#), [Github Pages](#)

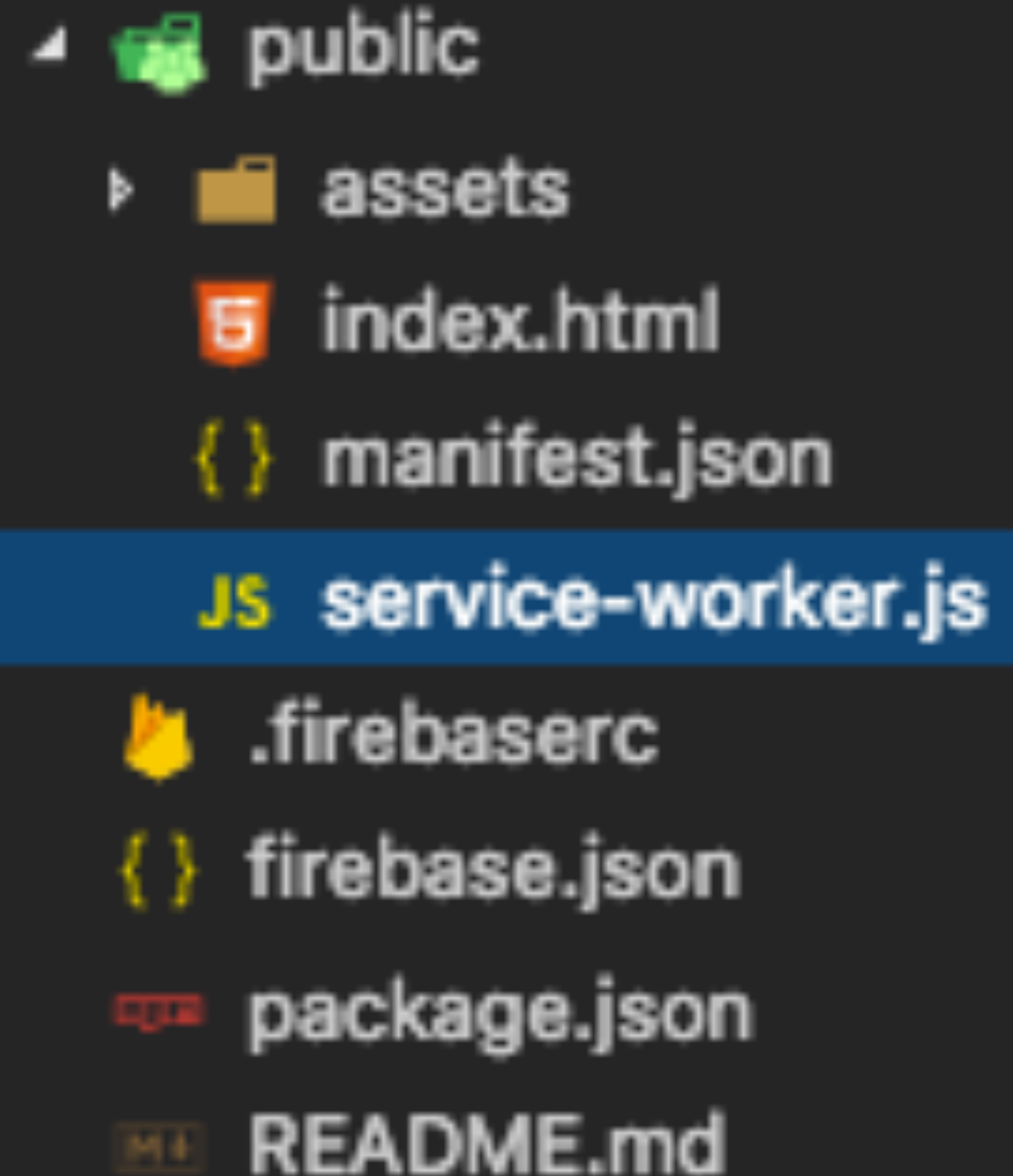
SERVICE WORKERS

---

REGISTER

# REGISTER A SERVICE WORKER

- ▶ Create a service worker file

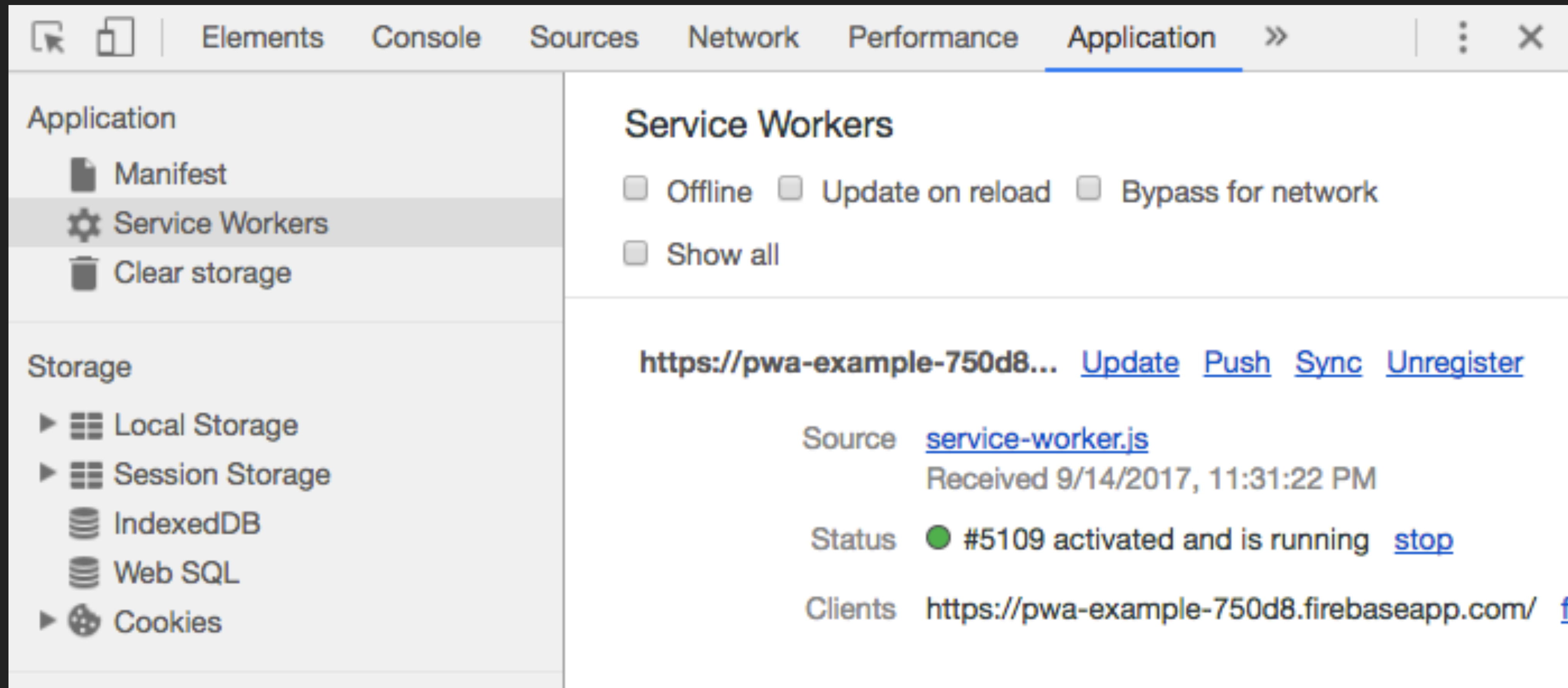


# REGISTER A SERVICE WORKER

► public/assets/app.js

```
if ('serviceWorker' in navigator) {  
  window.addEventListener('load', () => {  
    navigator.serviceWorker.register("/service-worker.js").then(register => {  
      console.log('Register succeeded in scope : ', register.scope)  
    })  
  })  
} else {  
  console.log('Service worker not support.')  
}
```

# REGISTER A SERVICE WORKER



# CHROME SERVICE WORKERS INSPECTOR

- ▶ <chrome://serviceworker-internals>
- ▶ <chrome://inspect/#service-workers>

SERVICE WORKERS

---

UPDATE

# UPDATE A SERVICE WORKER

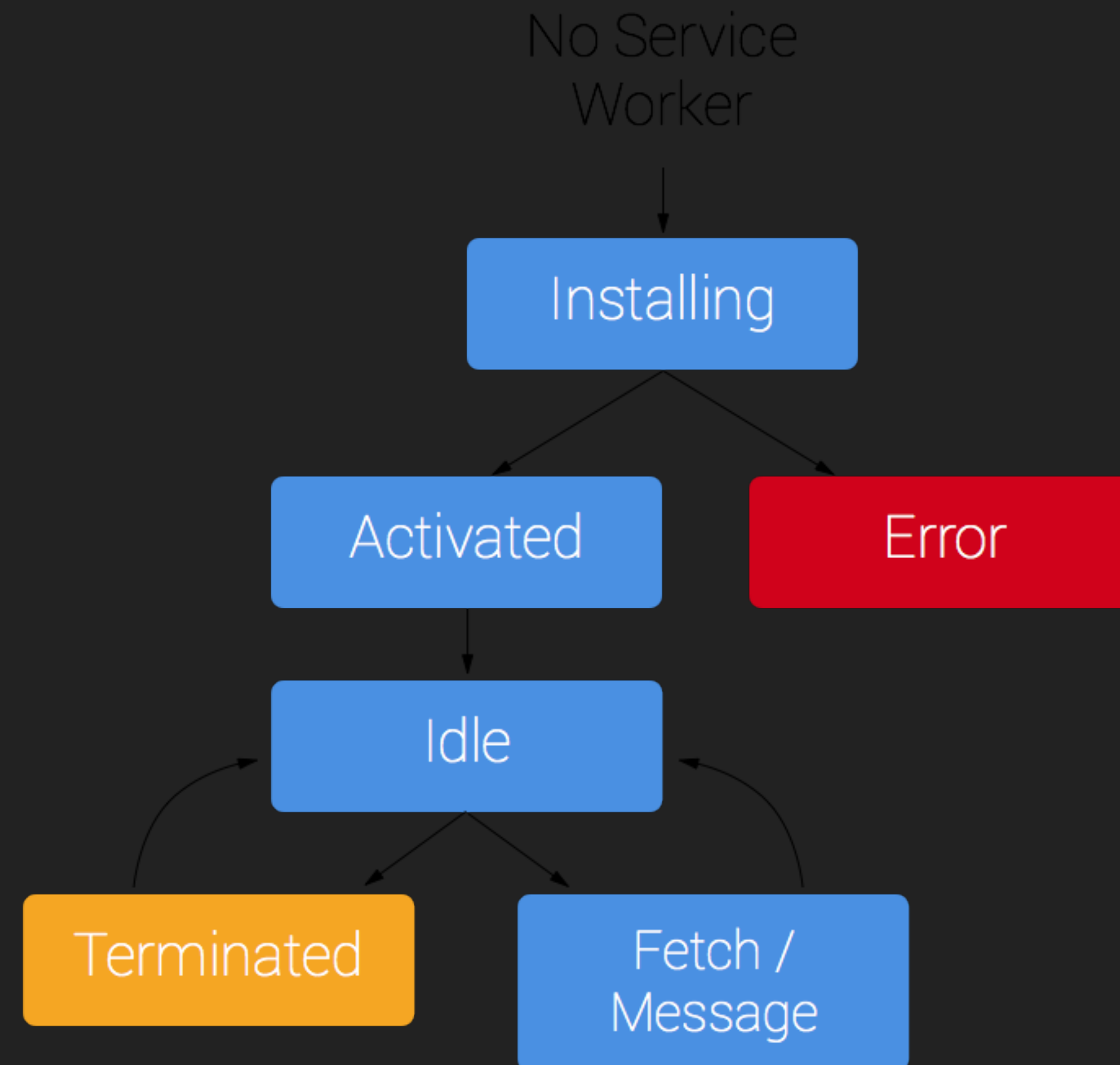
- ▶ If there is even a **byte's difference** in the service worker file compared to what it currently has, it considers it *new*



# UPDATE A SERVICE WORKER STEPS



# SERVICE WORKER LIFECYCLE

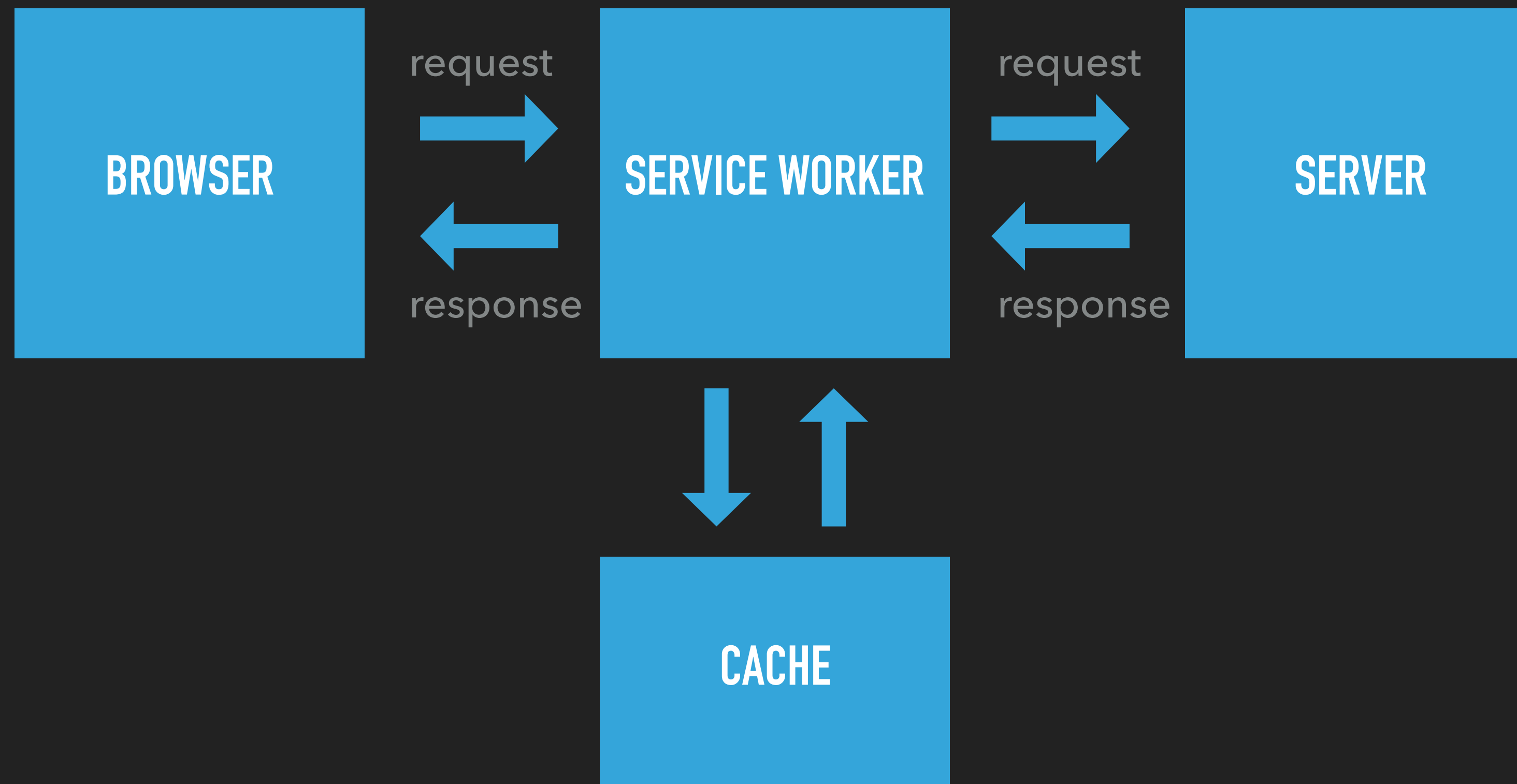


SERVICE WORKERS

---

OFFLINE MODE

# SERVICE WORKER AS NETWORK PROXY



# CACHE API

- ▶ Provides a storage mechanism for Request / Response object pairs that are cached

key	value
request	response

# CACHE STATIC ASSETS

## ► service-worker.js

```
const VERSION = '1'
const CACHE_KEY = `cache-v${VERSION}`
const assetsToCache = [
  '/assets/css/app.css',
  '/assets/js/app.js',
]

self.addEventListener('install', event => {
  event.waitUntil(
    caches.open(CACHE_KEY)
      .then(cache => cache.addAll(assetsToCache))
  )
})
```

# HANDLING FETCH EVENT

## ► service-worker.js

```
self.addEventListener('fetch', event => {  
  event.respondWith(  
    caches.match(event.request).then(cacheResp => {  
      return cacheResp || fetch(event.request).then(response => {  
        return caches.open(CACHE_KEY).then(cache => {  
          cache.put(event.request, response.clone())  
          return response  
        })  
      })  
    })  
  )  
})
```

# CLEANUP OLD CACHE

## ► service-worker.js

```
self.addEventListener('activate', event => {
  const cacheWhitelist = [CACHE_KEY]

  event.waitUntil(
    caches.keys().then(keyList => {
      return Promise.all(keyList.map(function(key) {
        if (cacheWhitelist.indexOf(key) === -1) {
          return caches.delete(key)
        }
      }))
    })
  )
  return self.clients.claim()
})
```



# OFFLINE COOKBOOK

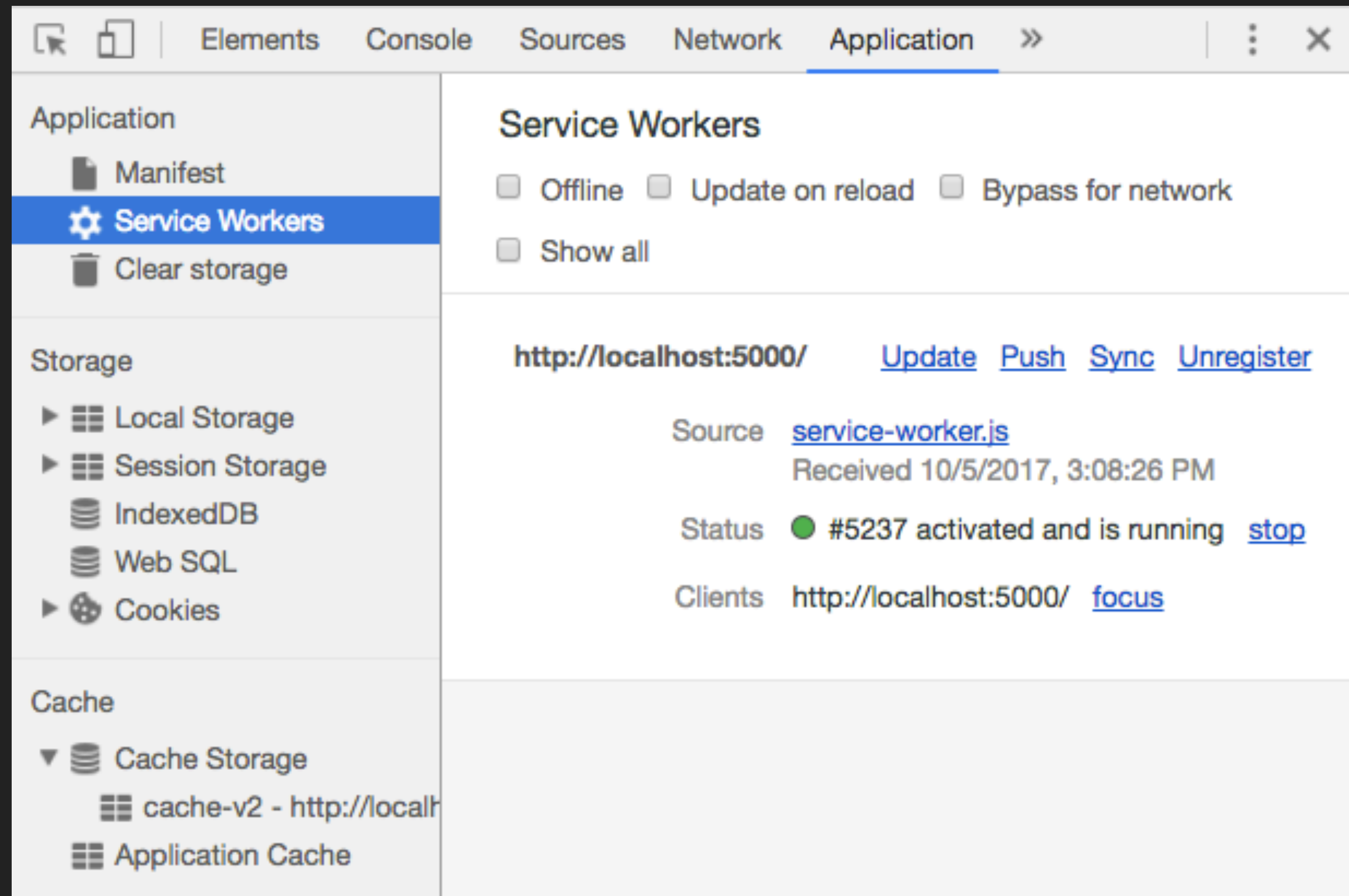
- ▶ Caching strategy - <https://jakearchibald.com/2014/offline-cookbook/#serving-suggestions-responding-to-requests>

SERVICE WORKERS

---

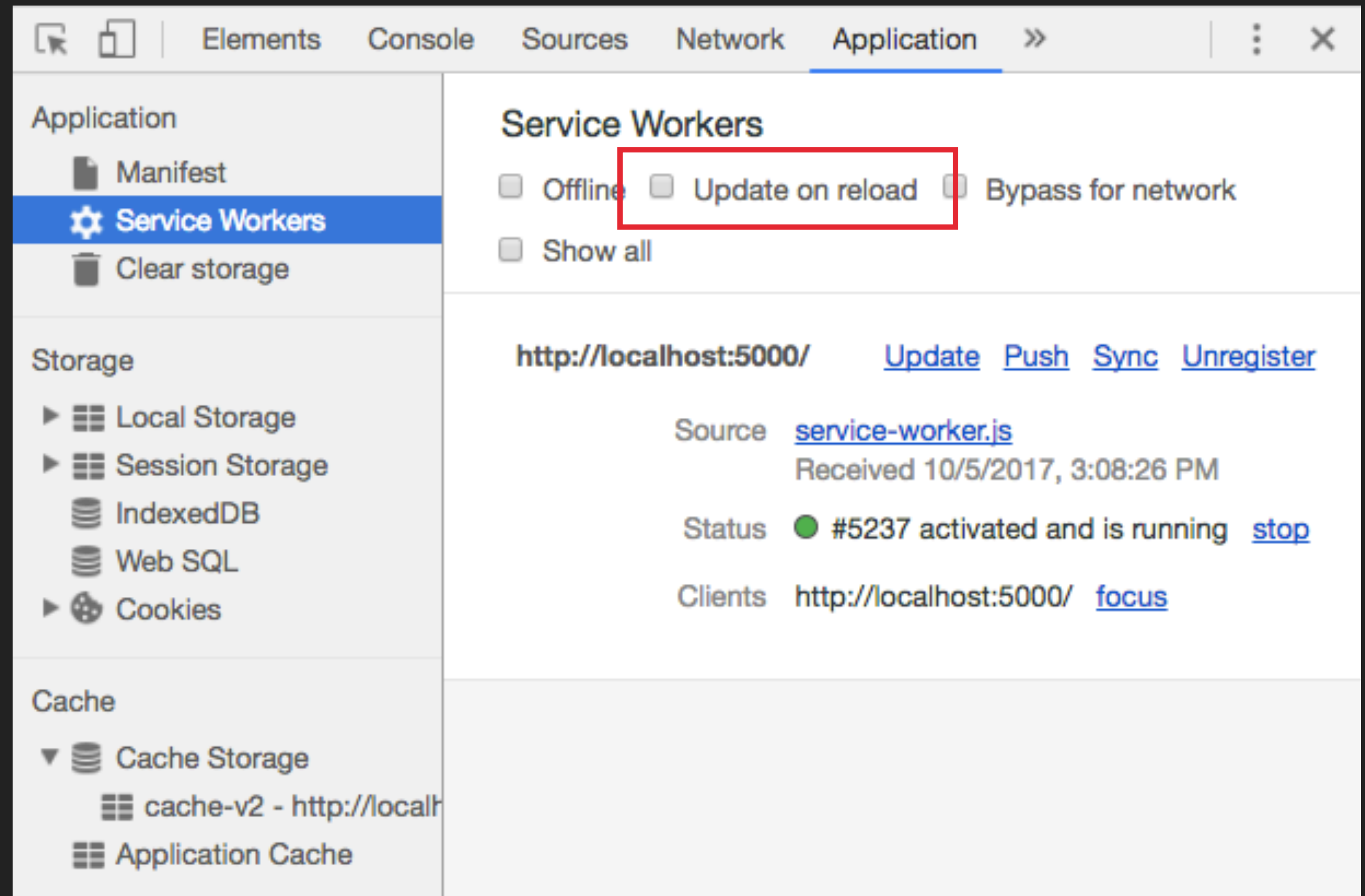
DEVELOPMENT TOOLS

# CHROME DEVELOPER TOOL



# UPDATE ON RELOAD

- ▶ Refetch service workers
- ▶ Install it as a new version even if it's byte-identical
- ▶ Skip the waiting phase so the new service worker activates





## CACHE STORAGE

Application				
<div>Manifest</div> <div>Service Workers</div> <div>Clear storage</div>				
Storage				
<div>Local Storage</div> <div>Session Storage</div> <div>IndexedDB</div> <div>Web SQL</div> <div>Cookies</div>				
Cache				
<div>Cache Storage</div> <div>cache-v2 - http</div> <div>Application Cach</div>				
#	Request	R...	Time Cached	
0	http://localhost:5000/	OK	10/5/2017, 3:02:50...	
1	http://localhost:5000/assets/css/app.css	OK	10/5/2017, 3:02:50...	
2	http://localhost:5000/assets/images/bac...	OK	10/5/2017, 3:02:50...	
3	http://localhost:5000/assets/images/bac...	OK	10/5/2017, 3:02:50...	
4	http://localhost:5000/assets/images/bac...	OK	10/5/2017, 3:02:50...	
5	http://localhost:5000/assets/images/icon...	OK	10/5/2017, 3:02:50...	
6	http://localhost:5000/assets/images/icon...	OK	10/5/2017, 3:02:50...	
7	http://localhost:5000/assets/images/icon...	OK	10/5/2017, 3:02:50...	
8	http://localhost:5000/assets/images/icon...	OK	10/5/2017, 3:02:50...	
9	http://localhost:5000/assets/images/icon...	OK	10/5/2017, 3:02:50...	
10	http://localhost:5000/assets/images/icon...	OK	10/5/2017, 3:02:50...	
11	http://localhost:5000/assets/images/icon...	OK	10/5/2017, 3:02:50...	
12	http://localhost:5000/assets/js/app.js	OK	10/5/2017, 3:02:50...	

# REPOSITORY

- ▶ <https://github.com/aofleejay/pwa-example/tree/step/2-service-workers>

## FEEDBACK

► <http://bit.ly/2xj66Ab>