



# PWA

## Progressive Web Application

Permet de sauvegarder côté client (dans le cache) une partie du site web à l'aide des "services workers".

Le service worker est une sorte de proxy.

## Exemple

Dans main.js (qui est appelé par une page html)

```
if (navigator.serviceWorker) {  
  navigator.serviceWorker.register('sw.js')  
  // Si le navigateur a un service worker, on installe 'sw.js'  
}
```

## Dans sw.js

stratégie cache first and then network

```
self.addEventListener('install', (event) => { // on écoute l'événement install du sw  
  console.log('installation de sw')  
  
  event.waitUntil( //s'assure que tous les événements se chargent dans le cache avant d'envoyer la suite  
    caches.open("cache01").then((cache) => {  
      return cache.addAll(["/", "/index.html", "/main.js"]); // charge dans le cache tes fichiers ( ! ne pas oublier le / avant le n  
    })  
  );  
})  
  
self.addEventListener('activate', (event) => { // on écoute l'événement activé du sw  
  console.log('activation de sw')  
})  
  
self.addEventListener("fetch", (event) => {  
  console.log("event fetch", event.request.url);  
  
  /*if (!navigator.onLine) { // si on est offline  
    event.respondWith(new Response("plus de connexion.. mode dégradé"));  
  }*/  
  
  //stratégie cache first and then network  
  event.respondWith(// si on est offline ça charge tous ce qui est dans le cache  
    caches.match(event.request).then((res) => {  
      return res;  
    })  
  );  
});
```

Stratégie network first and then cache callback

```

self.addEventListener('install', (event) => { // on écoute l'événement install du sw
  console.log('installation de sw')

  event.waitUntil( //s'assure que tous les événements se chargent dans le cache avant d'envoyer la suite
    caches.open("cache01").then((cache) => {
      return cache.addAll(["/", "/index.html", "/main.js"]); // charge dans le cache tes fichiers ( ! ne pas oublier le / avant le n
    })
  );
});

self.addEventListener('activate', (event) => { // on écoute l'événement activé du sw
  console.log('activation de sw')
});

self.addEventListener("fetch", (event) => {
  console.log("event fetch", event.request.url);

  event.respondWith(
    fetch(event.request).then( res => {
      caches.open("cache01").then(cache => cache.put(event.request, res))
      return res.clone();
    }).catch( err => {
      return caches.match(event.request);
    })
  )
});

```

## Dans le manifest.json

pour faire marcher son site comme une application sur android

Editeur de manifest en ligne :

- <https://www.simicart.com/manifest-generator.html/>
- <https://app-manifest.firebaseapp.com/>

link du manifest dans le html

```
<link rel="manifest" href="manifest.json" />
```

manifest.json

```

{
  "name": "tuvabien",
  "short_name": "tuv",
  "theme_color": "#2196f3",
  "background_color": "#2196f3",
  "display": "standalone",
  "scope": "/",
  "start_url": "/"
}

```