

L'Institut Supérieur des Technologies de l'Information & de la Communication

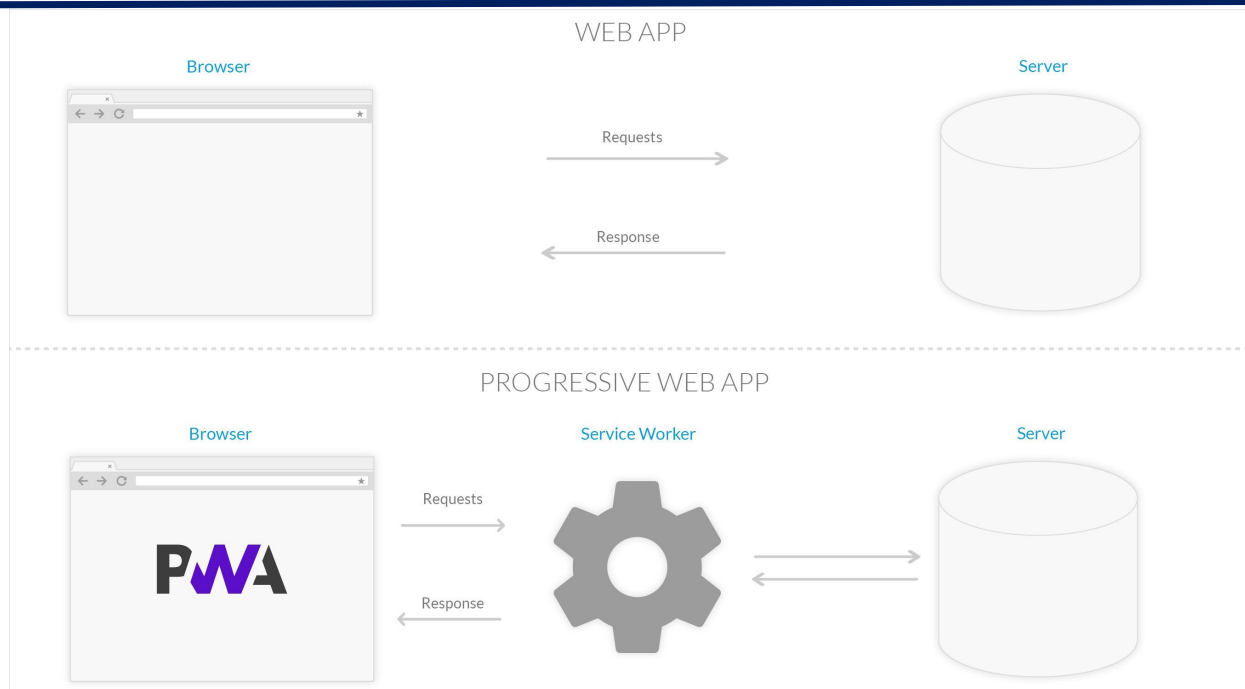
# Technologies & programmation web

## Partie Front End

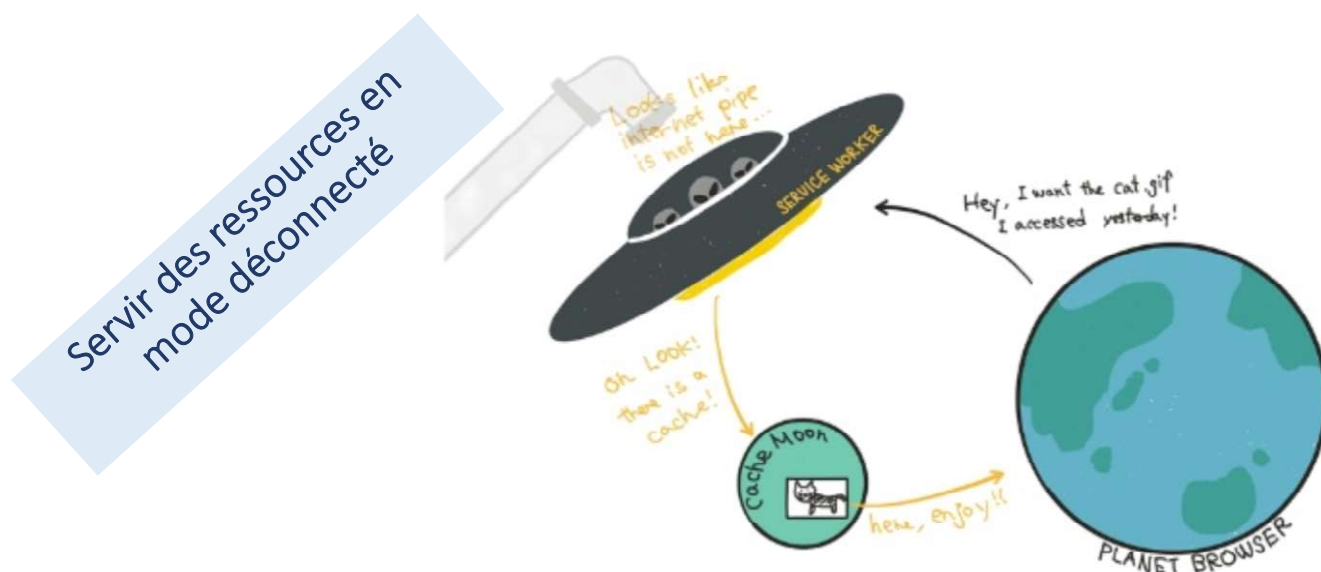
### Chapitre 2

### Application Web Progressive (PWA)

## Service Worker : C'est quoi?



# Service Worker : A quoi sert?



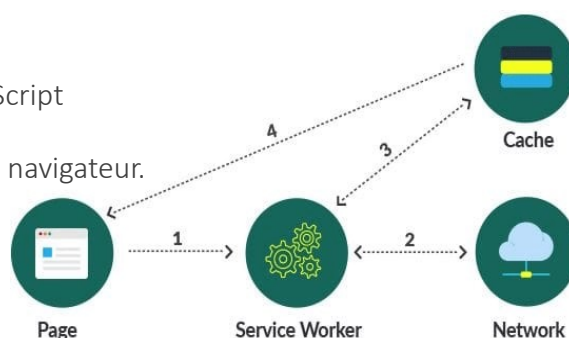
## Service Worker : Définition et Principe

### Définition:

- Un service worker prend souvent la forme d'un fichier JavaScript
- Il joue le rôle de **proxy** entre une application, le réseau et le navigateur.

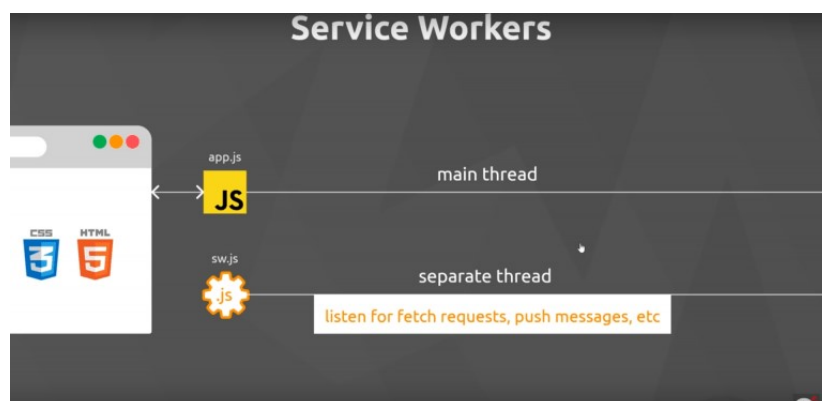
### Principe de fonctionnement:

Il permet de contrôler l'application web auquel il est associé, en interceptant les requêtes de ressources, et en mettant en cache ces ressources pour donner une maîtrise complète de la manière dont doit se comporter l'application en cas d'indisponibilité (problème réseau, problème au niveau du serveur, ...)



# Service Worker : Définition et Principe

- Il n'est pas lié au DOM du document
- Il s'exécute sur un thread distinct du thread principal du navigateur.
- > Il n'est pas bloquant



## Service Worker : Principe

The screenshot shows a web browser with a YouTube video player and the Chrome DevTools Application panel. The video is titled "C'est quoi une PROGRESSIVE WEB APP (PWA) ?" by WeBerger - Bryan BERGER. The Application panel shows the "Service Workers" section with a list of registered workers. The "Cache" section shows the "Cache d'application" for "https://www.youtube.com". The "En-têtes" (Headers) section shows the "Général" (General) tab with details for the request to "https://www.youtube.com/app\_shell\_home".

#	Nom	Typ...	Type...	Cont...	Heur...
0	/app_shell_home	basic	text/...	1,336	10/0...
1	/s/desktop/67548ecc/cssbin/...	basic	text/...	929	10/0...
2	/s/desktop/67548ecc/cssbin/...	basic	text/...	239	10/0...
3	/s/desktop/67548ecc/cssbin/...	basic	text/...	708	10/0...
4	/s/desktop/67548ecc/jsbin/c...	basic	text/...	1,16...	10/0...
5	/s/desktop/67548ecc/jsbin/d...	basic	text/...	5,243	10/0...
6	/s/desktop/67548ecc/jsbin/n...	basic	text/...	2,521	10/0...
7	/s/desktop/67548ecc/jsbin/sc...	basic	text/...	13,242	10/0...
8	/s/desktop/67548ecc/jsbin/s...	basic	text/...	15,145	10/0...
9	/s/desktop/67548ecc/jsbin/w...	basic	text/...	21,309	10/0...
10	/s/desktop/67548ecc/jsbin/W...	basic	text/...		

**En-têtes** Aperçu

**Général**

URL de la demande: https://www.youtube.com/app\_shell\_home

Méthode de demande: GET

Cote d'état: 200

**En-têtes de réponse**

alt-svc: h3-29=":443"; ma=2592000,h3-T051=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"

cache-control: no-cache, no-store, max-age=0, must-revalidate

content-encoding: br

content-type: text/html; charset=utf-8

Date: Wed, 10 Feb 2021 15:41:31 GMT

Nombre total d'entrées: 20

# Service Worker : Comment le déployer?

---

## 1- Vérifier si le navigateur supporte le ServiceWorker

```
if('serviceWorker' in navigator) {  
    // Supported ☐  
} else {  
    // Not supported ☐  
}
```

# Service Worker : Comment le déployer?

---

## 2- Enregistrer le ServiceWorker (cas du serviceWorker supporté)

Afin d'enregistrer le ServiceWorker:

- Le fichier du ServiceWorker doit être accessible
- Le fichier du ServiceWorker ne doit contenir que du code relatif au service worker
- Il faut

- ❖ attendre le chargement de la page
- ❖ ensuite transmettre le chemin du fichier du service worker à la méthode  
**navigator.serviceWorker.register ()**

# Service Worker : Comment le déployer?

## 2- Enregistrer le ServiceWorker (cas du serviceWorker supporté)

```

window.addEventListener('load', () => {
  if ('serviceWorker' in navigator) {
    // register service worker
    navigator.serviceWorker.register('/sw-worker.js').then(
      () => {
        console.log('SW registration successful ☐ ');
      },
      err => {
        console.error('SW registration failed ☐ ', err)
      }
    );
  } else {
    // Not supported ☐
  }
});

```

attendre le chargement de la page

nom du fichier du service worker

# Service Worker : Comment le déployer?

## 2- Enregistrer le ServiceWorker (cas du serviceWorker supporté)

### Remarques:

- Le service est enregistré avec une portée (scope)
- La portée (scope) du service worker détermine le chemin à partir duquel le service worker intercepte les requêtes.
- La portée par défaut est le chemin vers le script du service worker et s'étend sur tous ses sous-répertoires.
- Le service worker est fonctionnel via **https**. L'utilisation du protocole https garantit qu'il n'y a pas eu d'usurpation lors de l'installation du service worker. Si la page est servie en http; le service worker ne sera pas installé.



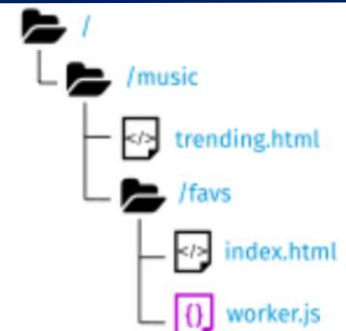
# Service Worker : Comment le déployer?

## 2- Enregistrer le ServiceWorker (cas du serviceWorker supporté)

### Exemple :

Supposons que worker.js est le fichier java script du service worker

- La portée par défaut du service est : .....
- La ressource que le service worker pourra servir est : .....
- La ressource que le service worker ne pourra pas servir est : .....



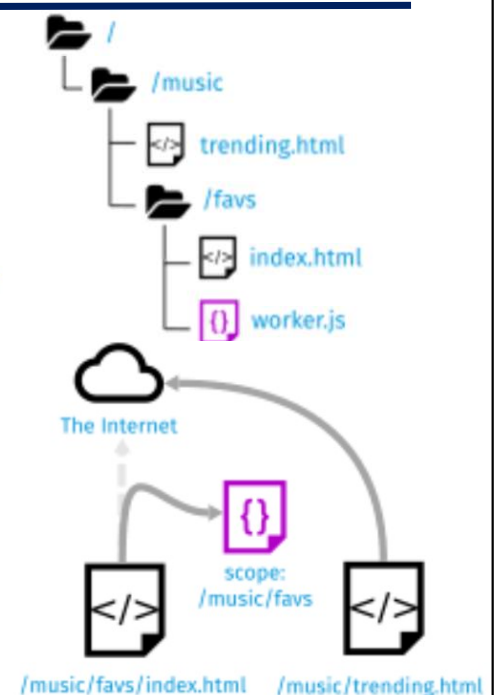
# Service Worker : Comment le déployer?

## 2- Enregistrer le ServiceWorker (cas du serviceWorker supporté)

### Exemple :

Supposons que worker.js est le fichier java script du service worker

- La portée par défaut du service est : `/music/favs/`
- La ressource que le service worker pourra servir est : `index.html`
- La ressource que le service worker ne pourra pas servir est : `trending.html`





## Service Worker : Son cycle de vie

---

Suite à son enregistrement, le service worker passe par un cycle de vie qui comprend les trois étapes suivantes:



Télécharger

Installer

Activer

## Service Worker : Son cycle de vie

---

### 1- Télécharger

Le service worker est immédiatement téléchargé lorsqu'un utilisateur accède pour la première fois à une page ou à un site contrôlé par un service worker.

### 2- Installer

Une fois téléchargé par le navigateur, le service worker peut être installé grâce à l'évènement `install`. Cet évènement sera déclenché si le navigateur considère le service worker comme nouveau (première installation ou une mise à jour).

L'évènement `install` est utilisé pour définir les instructions qui s'exécuteront durant l'installation, principalement les instructions de mise en cache des ressources.

## Service Worker : Son cycle de vie

---

### Exemple :

```
var cacheName = 'first-test';
var filesToCache = ['/FirstApplication/index.html'];

self.addEventListener('install', function (e) {
  console.log('[ServiceWorker] Install');
  e.waitUntil(
    caches.open(cacheName).then(function (cache) {
      console.log('[ServiceWorker] Caching app files', );
      return cache.addAll(filesToCache);
    })
  );
});
```

## Service Worker : Son cycle de vie

---

### 3- Activer

Suite à la première installation d'un nouveau service worker (intallation avec succès), on peut passer à l'étape d'après : l'activation du service worker grâce à l'événement **activate**

Cet événement permet éventuellement de nettoyer les vieux caches associés avec la version précédente du service worker.



## Service Worker : Son cycle de vie

---

```
var cacheName = 'first-test';
var filesToCache = ['/FirstApplication/index.html'];

self.addEventListener('activate', function (e) {
  console.log('[ServiceWorker] Activate');
  e.waitUntil(
    caches.keys().then(function (keyList) {
      return Promise.all(keyList.map(function (key) {
        if (key !== cacheName) {
          console.log('[ServiceWorker] Removing old cache', key);
          return caches.delete(key);
        }
      }));
    })
  );
  return self.clients.claim();
});
```

## Hébergement PWA :

---

- Hébergement local sur un serveur web local
  - Hébergement sur Tomcat
  - Test du service worker
  - Test du stockage en cache
- Hébergement distant sur un serveur d'hébergement en ligne
  - Hébergement sur Firebase (à voir en TP)

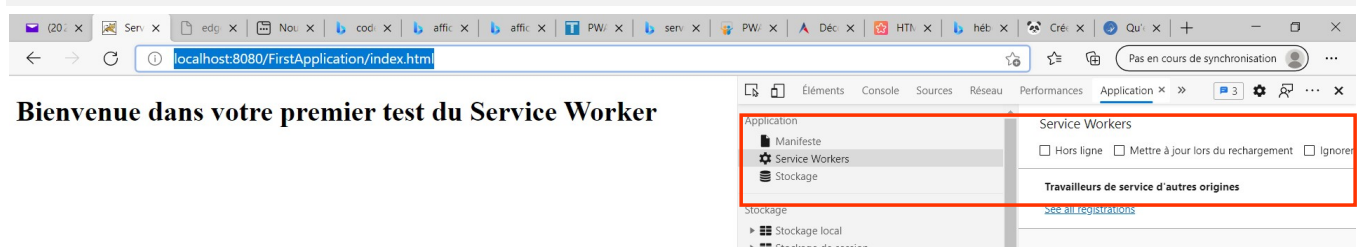
# Hébergement PWA : Local sur un serveur web

- Pré-requis : Installation du Serveur Tomcat
- Création d'un répertoire 'FirstApplication' contenant :
  - Dossier : src
  - Fichiers html basiques : index.html , message.html
  - Fichiers javascript : istic-sw.js
- Placer le répertoire 'FirstApplication' sous le serveur Tomcat au niveau du dossier webapps (dossier du serveur tomcat contenant toutes les applications web à déployer sur le serveur)
- Modifier les fichiers index.html et message.html de sorte qu'ils affichent des messages différents
- Démarrer Tomcat
- Accéder à l'adresse <http://localhost:8080/FirstApplication/index.html>

# Hébergement PWA : Local sur un serveur web

- index.html

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Service Worker- Test APPLICATION</title>
  </head>
  <body>
    <h1>Bienvenue dans votre premier test du Service Worker</h1>
  </body>
</html>
```



# Hébergement PWA : Local sur un serveur web

- Au niveau du fichier index.html ajouter un script qui permet d'enregistrer le service worker istic-sw.js

```
<script type="text/javascript">
  if ('serviceWorker' in navigator) {
    window.addEventListener('load', function() {
      navigator.serviceWorker.register('/FirstApplication/istic-sw.js').then(function(registration) {
        // Registration was successful
        console.log('ServiceWorker registration successful with scope: ', registration.scope);
      }, function(err) {
        // registration failed :(
        console.log('ServiceWorker registration failed: ', err);
      });
    });
  }
</script>
```

# Hébergement PWA : Local sur un serveur web

- Actualiser la page dans le navigateur

Bienvenue dans votre premier test du Service Worker

**Service Workers**

☐ Hors ligne ☐ Mettre à jour lors du rechargement ☐ Ignorer pour le réseau

**http://localhost:8080/FirstApplication/** Demandez réseau Mise à jour Annuler l'inscription

Source: istic-sw.js

Requ 11/02/2021 à 10:51:08

Statut: #26 activé et est en cours d'exécution [Arrêter](#)

Clients: http://localhost:8080/FirstApplication/index.html [mise au point](#)

Émission: Testez la notification Push de DevTools. [Émission](#)

Synchroniser: test-tag-from-devtools [Synchroniser](#)

Synchronisation périodique: test-tag-from-devtools [Synchronisation périodique](#)

Travailleurs de service d'autres origines

[See all registrations](#)

Console: Certains messages ont été déplacés vers le panneau problèmes. ServiceWorker registration successful with scope: http://localhost:8080/FirstApplication/index.html:20

# Hébergement PWA : Local sur un serveur web

- Modifier le service worker afin d'installer et activer le cache des ressources

```
var cacheName = 'first-testt';
var filesToCache = ['/FirstApplication/index.html'];

self.addEventListener('install', function (e) {
  console.log('[ServiceWorker] Install');
  e.waitUntil(
    caches.open(cacheName).then(function (cache) {
      console.log('[ServiceWorker] Caching app files', );
      return cache.addAll(filesToCache);
    })
  );
});
```

```
self.addEventListener('activate', function (e) {
  console.log('[ServiceWorker] Activate');
  e.waitUntil(
    caches.keys().then(function (keyList) {
      return Promise.all(keyList.map(function (key) {
        if (key !== cacheName) {
          console.log('[ServiceWorker] Removing old cache', key);
          return caches.delete(key);
        }
      }));
    })
  );
  return self.clients.claim();
});

self.addEventListener('fetch', function (e) {
  console.log('[ServiceWorker] Fetch', e.request.url);
  e.respondWith(
    caches.match(e.request).then(function (response) {
      return response || fetch(e.request);
    })
  );
});
```

# Hébergement PWA : Local sur un serveur web

- Modifier le service worker afin d'installer et activer le cache des ressources

Bienvenue dans votre premier test du Service Worker

Service Workers

☐ Hors ligne ☐ Mettre à jour lors du rechargement ☐ Ignorer pour le réseau

**http://localhost:8080/FirstA...** Demandez réseau Mise à jour Annuler l'inscription

Source [istic-sw.js](#)

Reçu 11/02/2021 à 11:04:42

Statut ● #31 activé et est en cours d'exécution [Arrêter](#)

Clients [http://localhost:8080/FirstApplication/index.html](#) [mise au point](#)

Émission Testez la notification Push de DevTools. Émission

Synchroniser test-tag-from-devtools Synchroniser

Synchronisation périodique test-tag-from-devtools Synchronisation périodique

Travailleurs de service d'autres origines

[See all registrations](#)

Console

Certains messages ont été déplacés vers le panneau problèmes. [Voir les problèmes](#)

[ServiceWorker] Install [istic-sw.js:5](#)

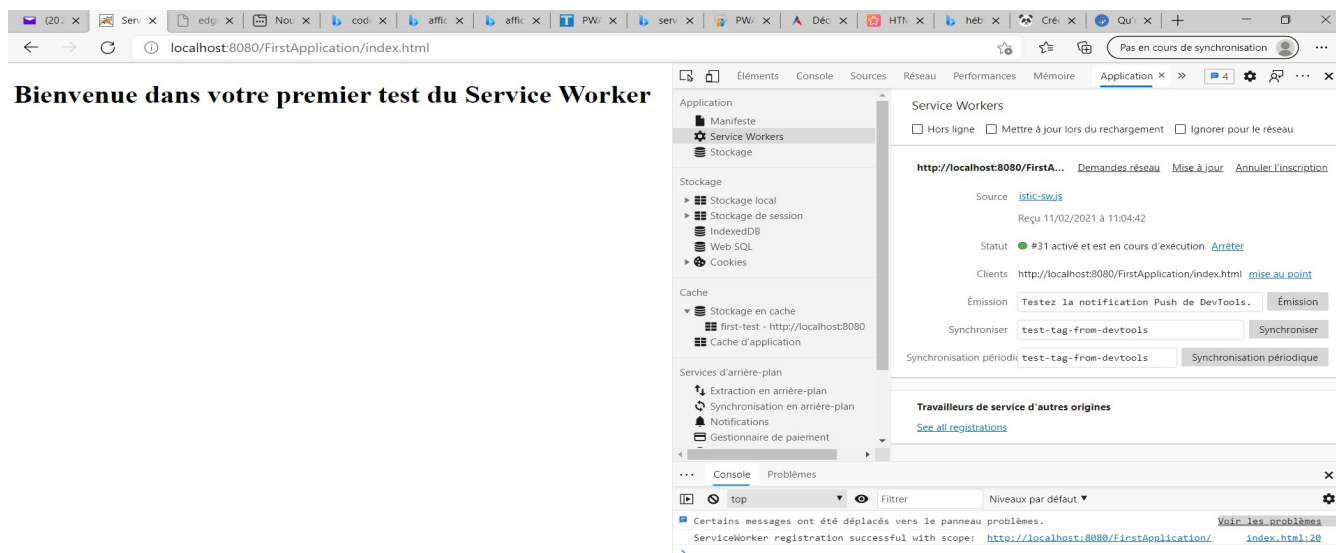
ServiceWorker registration successful with scope: [http://localhost:8080/FirstApplication/](#) [index.html:20](#)

[ServiceWorker] Caching app files [istic-sw.js:8](#)

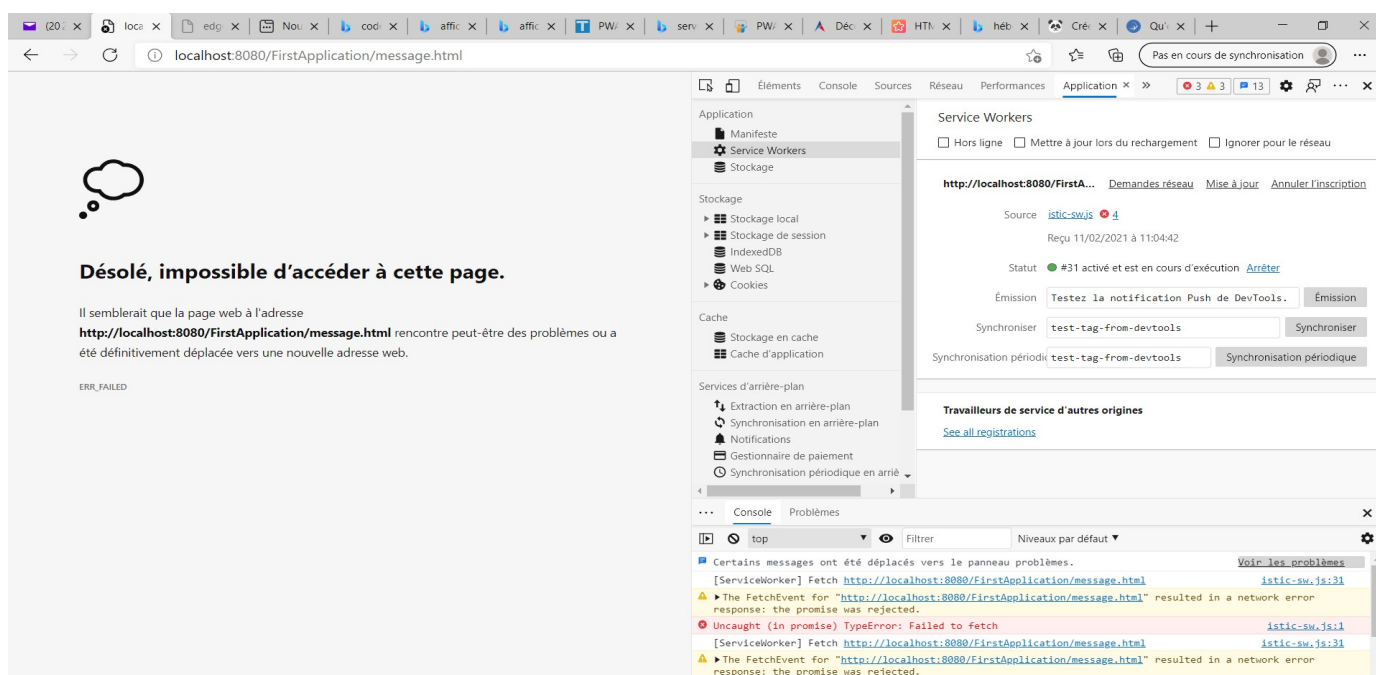
[ServiceWorker] Activate [istic-sw.js:16](#)

# Hébergement PWA : Local sur un serveur web

- Arrêter le serveur Tomcat
- Accéder au deux ressources index.html et message.html



# Hébergement PWA : Local sur un serveur web



## Hébergement PWA : Local sur un serveur web

---

- Modifier le fichier istic-sw.js afin d'activer le cache pour la ressource message.html
- Démarrer le serveur et faire les tests nécessaires
- Déplacer le service worker vers le dossier **src** et vérifier si vous pouvez toujours accéder aux ressources index.html et message.html

ENJOY :)