PWA

B2

Décembre 2025 - Nicolas-Paul ALBRECHT

ICE BREAKER

OBJECTIFS

B2 - PWA

- Connaitre les caractéristiques d'une PWA
- Comprendre les avantages et inconvénients
- Estimer quand avoir recours à une PWA

QU'EST-CE QU'UNE PWA ?

DÉFINITION

Progressive Web App

- Application Web améliorée pouvant être installée comme une application mobile.
- Comportement similaire à une application native.
- Possibilité de fonctionner hors-ligne.
- Peut envoyer des notifications push.

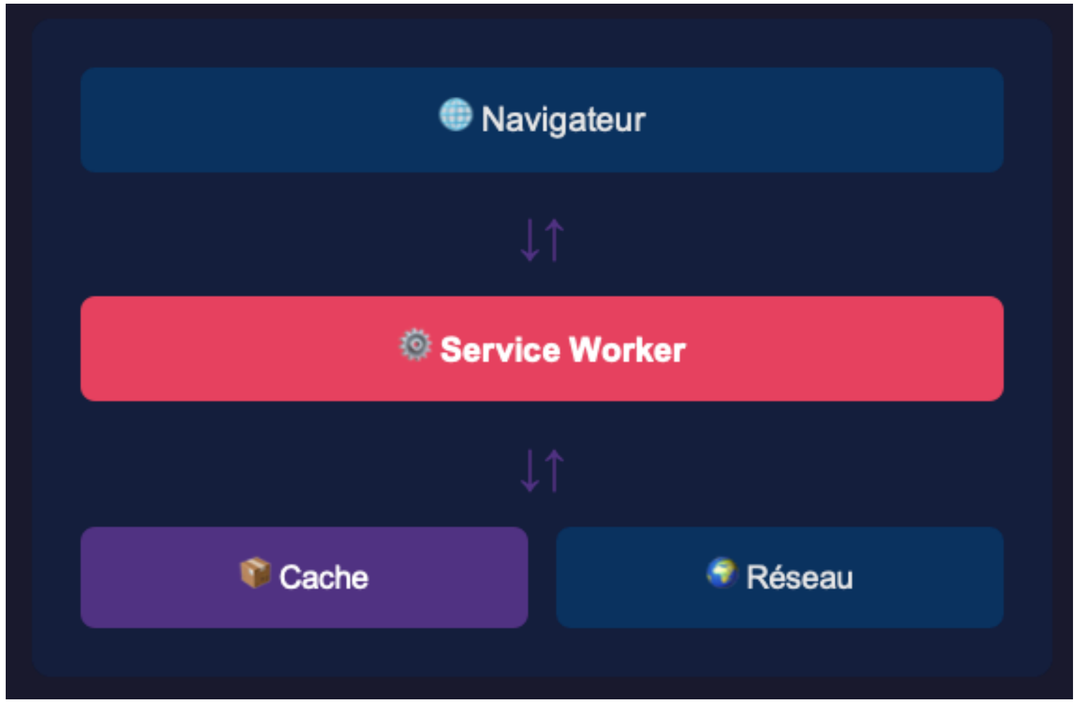
ARCHITECTURE

LES 3 CARACTERISTIQUES TECHNIQUES

SERVICE WORKER

- Script JS d’arrière plan
- Intercepte les requêtes
- Mise en cache intelligente des données
- Permet l’utilisation hors-ligne

SERVICE WORKER



Cycle de vie

- *Register*: Inscription du SW
- *Install:* Mise en cache initiale
- *Activate*: Prise de contrôle
- *Fetch*: Interception requêtes

SERVICE WORKER

1. CONFIGURATION

```
const CACHE_NAME ='mon-app-v1';
const ASSETS = [
'/','/index.html','/style.css',
'/app.js','/icon-192.png'
];
```

Seule partie à modifier.
CACHE_NAME: nom unique + version
ASSETS: tous les fichiers nécessaires au mode offline

2. INSTALL

```
self.addEventListener('install', e => {
e.waitUntil(
caches.open(CACHE_NAME)
.then(c => c.addAll(ASSETS))
);
});
```

Ouvre un espace de stockage et y copie tous les fichiers listés dans ASSETS
→ fonctionnement hors-ligne dès la 2° visite.

SERVICE WORKER

3. ACTIVATE

```
self.addEventListener('activate', e => {
e.waitUntil(caches.keys().then(keys =>
Promise.all(keys
.filter(k => k !==CACHE_NAME)
.map(k => caches.delete(k)))));
});
```

Quand CACHE_NAME change, supprimer automatiquement l'ancien cache. Libère l'espace et force la MAJ.

4. FETCH - CACHE-FIRST

```
self.addEventListener('fetch', e => {
e.respondWith(
caches.match(e.request)
.then(r => r ||fetch(e.request))
);
});
```

Interception de chaque requête

STRATÉGIES DE CACHE

2 types de mise en cache dans une PWA :

CACHE-FIRST

Idéal pour des fichiers statiques (HTML / CSS/ JS), les images & icônes, les polices de caractères ou les assets qui changet rarement.

NETWORK-FIRST

Idéal pour l'utilisation de données API, le contenu fréquemment mis à jour, les profils utilisateurs et les données en temps réel.

MANIFEST.JSON

- Fichier de configuration de la PWA
- Contient les informations comme : le nom, l'icône, les couleurs, le mode d'affichage, etc.

MANIFEST.JSON

Propriétés clés

- *display*: standalone | fullscreen | minimal-ui | browser
- *icons*. Tailles recommandées 192px et 512px.
- *theme_color*: Couleur de la barre de statut mobile
- *start_url*: Page de démarrage de l'app

HTTPS

- Contrainte de sécurité obligatoire dans une PWA
- Connexion chiffrée → protection des données
- Sans le protocole, la plupart des navigateurs bloqueront l'exécution des Services Workers.

TP N° 1

TP N° 1

- Prendre son application préféré
- Lister s'ils ont un site Web, une app native ou une PWA.
- Restituer à la classe + lister un élément de découverte sur l'application choisie.

AVANTAGES / INCONVENIENTS

AVANTAGES

- Pas d’obligations de publier dans les stores Android & Apple.
- Mises à jour automatique sans actions de l’utilisateur.
- Coût de développement fortement réduit.
- Application plus légère.
- Partage facile : un simple lien suffit.
- Fonctionnement hors-ligne.

INCONVENIENTS

- Accès aux API natives des téléphones limités. Ex: NFC, Bluetooth, Contacts, SMS,...
- Support IOS encore incomplet
- Consommation de la batterie plus important si l'application n'est pas optimisée.
- Visibilité réduite car absence des stores, moins de découvrabilité **SAUF si utilisation d'un PWA Wrapper.**
- Non adapté pour des applications complexes

AVANTAGES / INCONVENIENTS

Critère	Site Web	PWA	App Native
Installation	Aucune	Optionnelle	Store requis
Mode offline	❌ Non	✅ Oui	✅ Oui
Push notifs	❌ Non	⚠️ Partiel	✅ Oui
Accès matériel	Limité	⚠️ Moyen	✅ Complet
Coût dev	💰 Bas	💰 Moyen	💰💰💰 Élevé
Performance	Variable	⚠️ Bonne	✅ Excellente

QUAND CHOISIR UNE PWA ?

BON CHOIX POUR :

- Budget limité, besoin multiplateforme
- Contenu informaationnel / e-commerce
- Besoin hors-ligne ponctuel
- Mises à jour fréquentes
- Besoin multiplateforme rapide

PAS IDÉAL POUR :

- Jeux 3D / apps graphiques
- Accès Bluetooth/NFC avancé
- Intégration système profondes
- Monétisation via les stores (app payante)
- Cibler uniquement un système (Android ou Apple)

TP N° 2

TP N° 2

- Groupes de 4 personnes
- Un sujet va être donné, vous devez dire si le besoin doit être construit par :
  - Une PWA
  - Une App Native
  - Un site Web
- Restitution avec débat

POURQUOI CERTAINES ENTREPRISES CONSTRUISENT-ELLES UNE PWA ET UNE APP NATIVE ,

EXEMPLE DE UBER

- La PWA fait 600ko environ et est utilisée dans les pays à très faibles connexions Internet (Inde, Afrique, Asie du Sud-Est). L'application est capable de fonctionner en 2G !
- La PWA se concentre sur les fonctions de base :
  - localisation
  - recherche
  - commande de course

EXEMPLE DE UBER

- Les applications natives (Kotlin pour Android et Swift pour IOS) ont été construites pour améliorer l'expérience utilisateur.

- Les interactions sont plus poussées, l'intégration au système est meilleure, paiement en NFC, notifications riches, ...

La PWA permet donc, dans ce cas, de tester des marchés émergents rapidement avant d'offrir une vraie expérience via l'application si c'est compatible.

ALLER PLUS LOIN

NOTIFICATIONS

- 3 états de permission pour une notification :
  - default: pas encore demandé, popup apparaîtra
  - granted: Autorisé, la notification s'affichera
  - denied: Refusé, impossible d'afficher des notifications.

REGLES

- Il est important de demander l'autorisation après une action utilisateur, en expliquant pourquoi avoir besoin d'accéder aux notifications.
- Si déjà "denied", ne plus redemander.
- Ne pas spammer l'utilisateur.

NOTIFICATIONS

- 2 types de notifications :
  - Locales: déclenchées par le code de l'application, quand elle est en cours de fonctionnement.
  - Push: envoyées par un serveur même lorsque l'application est fermée. Nécessite un serveur backend.

NOTIFICATIONS

```
btn.addEventListener('click',async () => {
const perm =await Notification.requestPermission();
// perm = 'granted' | 'denied' | 'default'
});
```

```
if (Notification.permission ==='granted') {
// ✅ OK pour envoyer
}else if (Notification.permission ==='denied') {
// ❌ Bloqué définitivement
}
```

```
newNotification('Alerte Météo', {
body:'Pluie prévue dans 2h !',
icon:'icon-192.png',
tag:'meteo-alert'// Évite doublons
});
```

body : texte | **icon**: image 192px | **tag** : ID unique | **silent :** sans son (boolean) | **requireInteraction :** reste affichée (boolean)

TP N° 3

TP N° 3

- Construction d'une PWA de météo
- La PWA devra contenir uniquement: un champ de recherche pour saisir la ville et afficher les résultats.
- Elle devra envoyer une notification si la ville choisie va avoir de la pluie dans 4 prochaines heures ou si la température va dépasser les 10°
- Utilisation de l'API Open-Meteo
- BONUS :
  - Un dark mode de l'application
  - Sauvegarder dans le local storage des villes favorites

TIPS

- Pour pouvoir installer le site sur mobile, utiliser Github Pages.
- Documentation: https://open-meteo.com/en/docs
- Icône de l'application et SW fournis

CHALLENGE

Obtenir + de 90% sur sa PWA avec Light House

MERCI