DÉVELOPPEMENT MOBILE

PWA : LES « PROGRESSIVE WEB APPS »

DÉFINITION ? UNE PWA C'EST QUOI…?

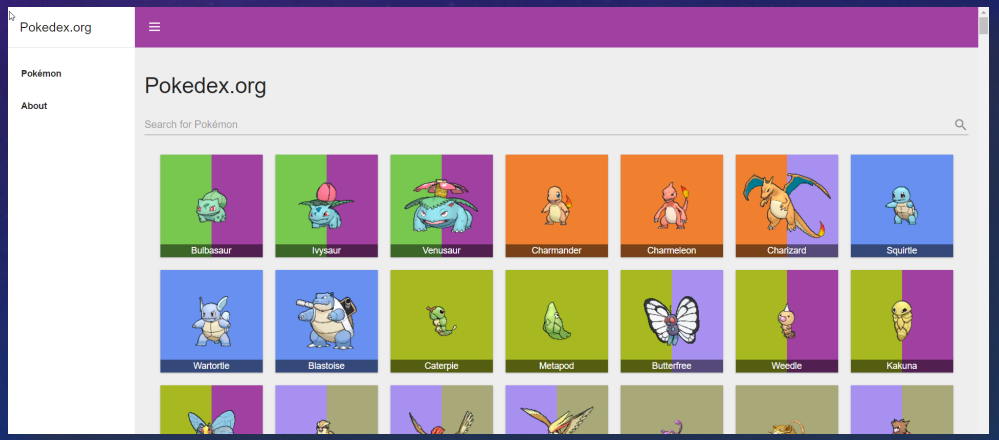
« Un ensemble de technologies et de bonnes pratiques qui permettent d'adapter un **service web fourni à l'utilisateur** pour le rendre accessible au delà des limites traditionnelles du navigateur et, notamment mais pas uniquement, dans des conditions sous-optimales d'accès au réseau (incluant le hors-ligne). »
(source: http://blog.clever-age.com)

- C'est un site web construit avec les technologies web mais qui se comporte et ressemble à une application mobile… Elles ont d'ailleurs des caractéristiques fondamentales bien précises.

- Un exemple: (https://pwa.rocks/)

http://devdocs.io/

https://www.pokedex.org/



10 CARACTÉRISTIQUES

« D'ici peu, l'utilisateur aura la possibilité d'installer une PWA sur son écran d'accueil, de la même manière qu'une application (avec empaquetage des données, présence dans la liste des applications, informations sur la consommation de mémoire et de batterie…). Ces innovations ont le potentiel de changer complètement la manière dont les internautes consomment le Web et les applications sur mobiles. » (source: http://blog.clever-age.com)

- Progressive, Discoverable, Linkable, Responsive, App-Like, Connectivity-independent, Re-engageable, Installable, Fresh, Safe (cf. Google)

1 - PROGRESSIVE

- Doit fonctionner sur tous les types de périphériques
  et
- S’améliorer progressivement en tirant partie des fonctionnalités disponibles sur les périphériques et navigateurs des utilisateurs

2 – DISCOVERABLE (RÉFÉRENCÉE)

- C’est un site web donc elle est facilement trouvable via les moteurs de recherche.
  - Gros avantage par rapport aux applications natives…

3 – LINKABLE (PARTAGEABLE)

- Pour se repérer dans un site et retrouver l'endroit précisément où on était, on utilise une URI (un lien): c'est l'état courant de l'application… on peut donc retenir, retrouver et recharger un état donné (favori ou partage) dans une PWA.

4 - RESPONSIVE

- L’interface utilisateur doit s’adapter à la forme et à la taille d’écran des périphériques sur lesquels elle est affichée.

5 – APP-LIKE (INTERACTIONS AUX ALLURES NATIVES)

- Une PWA doit avoir l'aspect d'une application native

  et

- Etre construite de manière à limiter le nombre de rechargement de pages.

6 – CONNECTIVITY-INDEPENDENT (INDÉPENDANT DU RESEAU)

- Elle doit fonctionner même en condition de faible connectivité réseau voire en mode offline complet

7 – RE-ENGAGEABLE (FAVORISE LE RE-ENGAGEMENT)

- Les applications mobiles sont plus réutilisées par les utilisateurs: En s'appuyant sur les notifications de PUSH, les Progressive Web Apps doivent obtenir le même résultat.

8 - INSTALLABLE

- Une PWA doit pouvoir être installée sur l'écran d'accueil d'un périphérique afin d'être plus facilement accessible.
  - Avantage par rapport aux sites web classiques

9 – FRESH (A JOUR)

- Un nouveau contenu publié au sein d'une PWA doit être accessible instantanément pour l'ensemble des utilisateurs connectés à Internet.

10 – SAFE (SECURISEE)

- Une PWA se doit impérativement d'être hébergée en HTTPS (pour éviter les attaques de type interception).
  - Voir les Service Workers (https://developer.mozilla.org/fr/docs/Web/API/Service_Worker_API).

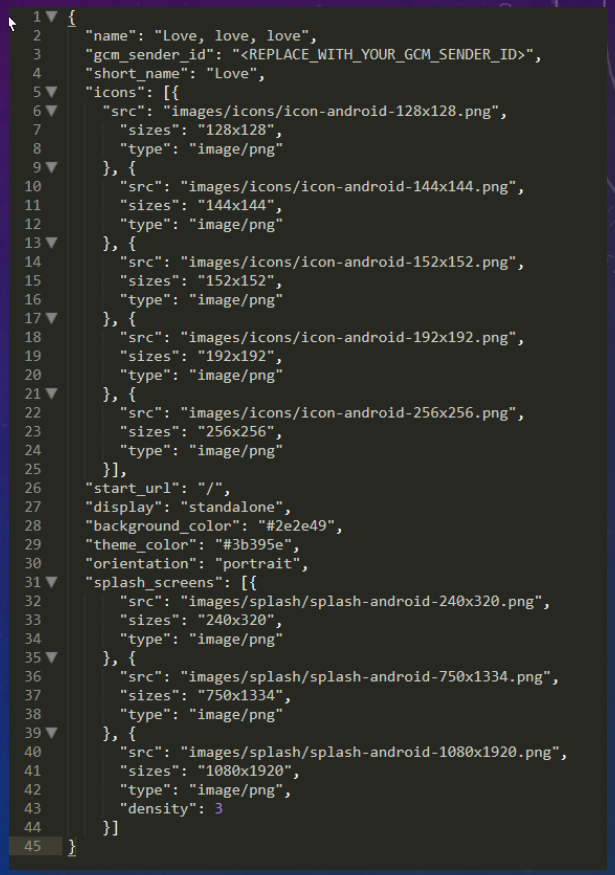
POURQUOI LES PWA?

- Constat: 20% de pertes d'utilisateurs à chaque fois qu'on ajoute une étape entre le premier contact visuel et le premier lancement effectif (recherche AppStore puis téléchargement, puis installée puis lancée).
  - ➢ Là, utilisation directe puis au second lancement : proposition d'installation et expérience plein écran!

TECHNIQUEMENT ? WEB APP MANIFEST

- Au-delà du HTML 5, CSS & JS pour le site… Web App Manifest:
  - Pour rendre l'application web plus proche d'une application en termes d'aspects ou de comportements avec la possibilité de la rajouter à l'écran d'accueil d'un smartphone.
  - C'est un fichier JSON normalisé par le W3C.

Exemple: https://github.com/MadeOnMars/lovelovelove



```
{
  "name": "Love, love, love",
  "gcm_sender_id": "<REPLACE_WITH_YOUR_GCM_SENDER_ID>",
  "short_name": "Love",
  "icons": [{
    "src": "images/icons/icon-android-128x128.png",
      "sizes": "128x128",
      "type": "image/png"
    }, {
      "src": "images/icons/icon-android-144x144.png",
      "sizes": "144x144",
      "type": "image/png"
    }, {
      "src": "images/icons/icon-android-152x152.png",
      "sizes": "152x152",
      "type": "image/png"
    }, {
      "src": "images/icons/icon-android-192x192.png",
      "sizes": "192x192",
      "type": "image/png"
    }, {
      "src": "images/icons/icon-android-256x256.png",
      "sizes": "256x256",
      "type": "image/png"
    }],
  "start_url": "/",
  "display": "standalone",
  "background_color": "#2e2e49",
  "theme_color": "#3b395e",
  "orientation": "portrait",
  "splash_screens": [{
      "src": "images/splash/splash-android-240x320.png",
      "sizes": "240x320",
      "type": "image/png"
    }, {
      "src": "images/splash/splash-android-750x1334.png",
      "sizes": "750x1334",
      "type": "image/png"
    }, {
      "src": "images/splash/splash-android-1080x1920.png",
      "sizes": "1080x1920",
      "type": "image/png",
      "density": 3
    }]
}
```

WEB APP MANIFEST

- C'est pour l'instant bien géré par Google Chrome; il doit comporter :
  - Le paramètre short_name (nom de l'application dans l'écran d'accueil)
  - Le paramètre name (nom utilisé dans la bannière d'installation)
  - Une image d'icône (144x144 au format png)
  - Le paramètre d'URL de démarrage start-url doit être spécifié
- En outre
  - Il doit avoir un service worker enregistré qui utilise HTTPS
  - Etre visité au moins 2 fois, avec au moins 5 min. entre chaque visite.

Pour le charger (dans index.html): < link rel =''manifest'' href='./manifest.json''>

SERVICE WORKERS

- Grâce à un script Javascript, exécuté en arrière plan lorsqu'un visiteur va visiter la PWA:
  - Il ne peut pas accéder au DOM mais peut communiquer avec les pages via la méthode postMessage;
  - Il peut intercepter les requêtes réseaux et ainsi modifier le comportement de ces dernières.
  - Supportés par Chrome, Firefox et Opéra. Edge en dev. Safari envisagé…
- *Ex: Permet ainsi d'afficher les données récupérées précédemment (via IndexedDB), une fois connecté on récupère les dernières données.*

NB: Un serveur HTTPS nécessite des certificats (cf. l'initiative https://letsencrypt.org et l'outil Certbot)

NOTIFICATION DE PUSH

- Deux API utilisables : Push API et Notification API
  - Le push permet au serveur de communiquer avec l'application (en fait le SW - service worker)
  - La notification permet au SW d'afficher la notification

NB: Services qui nécessitent des clés d'utilisation auprès de google.