

Angular : PWA

Achref El Mouelhi

Docteur de l'université d'Aix-Marseille
Chercheur en programmation par contrainte (IA)
Ingénieur en génie logiciel

`elmouelhi.achref@gmail.com`



PWA : Progressive Web App

- Concept introduit par **Google** en 2015
- Utilisé pour désigner les applications web utilisant
 - une navigation sécurisée (**HTTPS**)
 - des services **workers**
 - un fichier **Manifest**

Angular

Pour ajouter un service worker à notre projet

```
ng add @angular/pwa
```

© Achref EL MOUELHI ©

Angular

Pour ajouter un service worker à notre projet

```
ng add @angular/pwa
```

Résultat

```
CREATE ngsw-config.json (631 bytes)
CREATE src/manifest.webmanifest (1340 bytes)
CREATE src/assets/icons/icon-128x128.png (1124 bytes)
CREATE src/assets/icons/icon-144x144.png (1291 bytes)
CREATE src/assets/icons/icon-152x152.png (1306 bytes)
CREATE src/assets/icons/icon-192x192.png (1654 bytes)
CREATE src/assets/icons/icon-384x384.png (3557 bytes)
CREATE src/assets/icons/icon-512x512.png (5008 bytes)
CREATE src/assets/icons/icon-72x72.png (711 bytes)
CREATE src/assets/icons/icon-96x96.png (857 bytes)
UPDATE angular.json (2951 bytes)
UPDATE package.json (1081 bytes)
UPDATE src/app/app.module.ts (739 bytes)
UPDATE src/index.html (475 bytes)
```

Explication

- Icône d'**Angular** générée en plusieurs résolutions.
- Deux fichiers créés
 - `manifest.webmanifest` : fichier responsable de l'apparence de l'application **PWA** lorsqu'elle s'ouvrira.
 - `ngsw-config.json` : fichier de configuration du service worker.
- Déclaration de `manifest.webmanifest` et `ngsw-config.json` d'`angular.json`
- Importation de `theServiceWorkerModule` dans `app.module.ts` pour l'enregistrement de `ngsw-config.json` (uniquement pour le mode production).
- Deux modifications dans `index.html`
 - référencement de `manifest.webmanifest`
 - ajout d'une balise `meta` pour `theme-color`

Angular

Nouveau contenu d'`index.html`

```
<!doctype html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>CoursAngular</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <link rel="manifest" href="manifest.webmanifest">
  <meta name="theme-color" content="#1976d2">
</head>

<body>
  <app-root></app-root>
  <noscript>Please enable JavaScript to continue using this application
  .</noscript>
</body>

</html>
```

Contenu de manifest.webmanifest

```
{
  "name": "cours-angular",
  "short_name": "cours-angular",
  "theme_color": "#1976d2",
  "background_color": "#fafafa",
  "display": "standalone",
  "scope": "./",
  "start_url": "./",
  "icons": [
    {
      "src": "assets/icons/icon-72x72.png",
      "sizes": "72x72",
      "type": "image/png",
      "purpose": "maskable any"
    },
  ]
}
```

Explication

- `name` : Le nom de l'application.
- `short_name` : Le nom court de l'application (utilisé lorsque l'espace d'affichage est limité).
- `start_url` : L'URL de (démarrage) la page d'accueil.
- `display` : Détermine comment l'application doit être affichée. "`standalone`" permet d'afficher l'application dans le navigateur ou dans une fenêtre distincte.
- `icons` : Une liste d'icônes utilisées pour représenter l'application sur différents appareils et plateformes.
- `theme_color` : La couleur d'accentuation de l'application.
- `background_color` : La couleur de fond de l'application.

Angular

Exécutons maintenant la commande suivante

```
ng build
```

© Achref EL MOUELHI ©

Exécutons maintenant la commande suivante

```
ng build
```

Remarques

- `ng serve` ne fonctionne pas avec les services workers
- Nous devons donc installer un serveur HTTP : `http-server`
(<https://www.npmjs.com/package/http-server>)

Angular

Exécutons maintenant la commande suivante

```
ng build
```

Remarques

- `ng serve` ne fonctionne pas avec les services workers
- Nous devons donc installer un serveur HTTP : `http-server`
(<https://www.npmjs.com/package/http-server>)

Pour installer `http-server`

```
npm install --global http-server
```

Exécutons la commande suivante pour lancer `http-server` **depuis le répertoire qui a été généré par** `ng build`

```
http-server -p 8080 -c-1 dist/cours-angular
```

© Achref EL MOUADJID

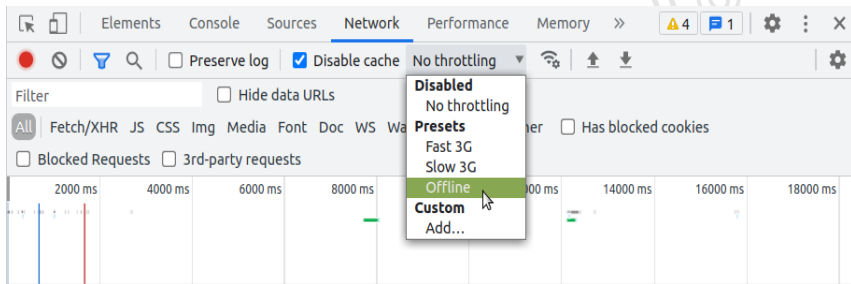
Exécutons la commande suivante pour lancer `http-server` **depuis le répertoire qui a été généré par** `ng build`

```
http-server -p 8080 -c-1 dist/cours-angular
```

Remarque

Pour tester, allez à `localhost:8080` et vérifiez que l'application se charge correctement.

Pour simuler un problème réseau, allons dans DevTools du navigateur et activons le mode offline



Allez à `localhost:8080` et vérifiez que l'application se charge correctement malgré l'absence d'une connexion réseau.

© Achref EL MOUËL

Allez à `localhost:8080` et vérifiez que l'application se charge correctement malgré l'absence d'une connexion réseau.

Remarque

Sans le service worker, on aurait eu le message `There is no Internet connection.`