18. Progressive Web Apps (PWAs) in React

Last updated by | Subramanya Dixit | May 5, 2025 at 7:59 PM GMT+5:30

📘 Why PWAs Are Important

Progressive Web Apps (PWAs) provide a native app-like experience in the browser. They are reliable, fast, and can be installed on a user's device, offering offline capabilities, push notifications, and more. PWAs combine the best of web and mobile apps, and React is a great framework to build them.

🧠 Key Concepts

✅ What is a PWA?

A Progressive Web App is a type of application that is:

- **Progressive**: Works for every user, regardless of browser choice.
- **Responsive**: Adapts to different screen sizes.
- **Offline First**: Uses Service Workers to work offline or on low-quality networks.
- **App-like**: Feels like an app to users, with smooth interactions and navigation.
- **Installable**: Can be added to the home screen of a device.
- **Push Notifications**: Can send updates to users even when they are not actively using the app.

✅ Setting Up a PWA with Create React App

Create React App (CRA) has built-in support for PWAs. To turn your app into a PWA, you need to enable service workers.

1. **Create a React App**:

   npx create-react-app my-pwa

2. **Enable Service Worker**:
   By default, CRA registers a service worker in the `src/index.js` file. You need to change `serviceWorker.unregister()` to `serviceWorker.register()` to enable it.

   import * as serviceWorkerRegistration from './serviceWorkerRegistration';

   serviceWorkerRegistration.register();

3. **Manifest File**:
   The manifest file contains metadata about the app (name, icons, theme color). It is typically found in the `public/manifest.json` file.

Example:

```
{
  "short_name": "React App",
  "name": "React Progressive Web App",
  "icons": [
    {
      "src": "icons/icon-192x192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "icons/icon-512x512.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ],
  "start_url": ".",
  "display": "standalone",
  "background_color": "#ffffff",
  "theme_color": "#000000"
}
```

✅ Service Workers

A **service worker** is a script that runs in the background and can intercept network requests to cache resources, making your app usable offline.
To add caching functionality, you can customize the `serviceWorker.js` file or use **Workbox**, a set of libraries that simplify service worker management.

```
// In serviceWorker.js
self.addEventListener('install', (event) => {
  event.waitUntil(
    caches.open('my-cache').then((cache) => {
      return cache.addAll(['/', '/index.html', '/static/js/main.js']);
    })
  );
});
```

✅ Offline Functionality

With a PWA, your app should be functional even when offline. This is handled by service workers that cache the necessary files.
To check if your app works offline, you can disable the network in your browser's developer tools and verify that your app loads correctly without an internet connection.

✅ Push Notifications

PWAs can send push notifications to users even when they are not actively using the app. This is accomplished by integrating with the **Push API** and **Notification API**.
Example of subscribing to push notifications:

```
// In serviceWorker.js
self.addEventListener('push', function (event) {
  const options = {
    body: event.data.text(),
    icon: '/images/icon.png',
    badge: '/images/badge.png',
  };
  event.waitUntil(self.registration.showNotification('New Notification', options));
});
```

💡 Guidelines

- **Manifest**: Always include a manifest file with metadata for your app.
- **Service Workers**: Use service workers to cache assets for offline functionality.
- **Responsiveness**: Ensure your app is responsive and can adapt to different screen sizes.
- **Push Notifications**: Integrate push notifications to keep users engaged even when they are not using the app.
- **Testing**: Test your PWA using Lighthouse (a tool in Chrome DevTools) to check its performance, accessibility, and PWA features.

📝 Practice Exercises

1. Convert an existing React app into a PWA.
2. Add offline functionality using service workers to your app.
3. Create a manifest file and test your app's installability.
4. Set up push notifications in your PWA app.

❓ Quiz Questions

1. What is the primary benefit of Progressive Web Apps (PWAs)?
a) They work only on mobile devices
b) They require internet connection to work
✅ **c) They are fast, reliable, and can work offline**
d) They cannot be installed on devices

---

2. Which file do you modify to enable PWA features in a React app created with Create React App (CRA)?
a) `index.html`
✅ **b)** `serviceWorker.js`
c) `manifest.json`
d) `App.js`

---

3. What does a service worker do in a Progressive Web App?
a) It caches assets and handles background tasks like push notifications
b) It provides the main UI elements of the app
✅ **c) It intercepts network requests to enable offline functionality**
d) It updates the app's code automatically
