

PWA

Progressive Web Apps

What makes a progressive web app?

— — —

- Close to native app experience
 - Designed and works like a native app
 - Ideally native app feature parity
 - Responsive to user interaction
 - Installable
- Caching
 - Fast loading
 - Can work offline
- Security
 - Must be served over HTTPS

Close Native App Experience

— — —

- Installable to Home Screen
 - Users can launch the site in “app mode” from Home Screen
- Smooth page transitions
 - Often a single page app
 - Load skeleton page
 - Show loading indicators
 - Clicks should not feel laggy
- Push notifications
- Background data sync

Caching

- Pages load quickly
 - Resources often load from cache first
- Pages can work offline
 - Users can see cached resources and data
- Service workers (more info later)
 - Specialized script that runs off of main thread
 - Can intercept every fetch request
 - Good for caching or preloading resource

Security

— — —

- HTTPS or bust

Technical Parts

— — —

- App manifest
 - Required for “Add to Home Screen”
 - App icons
 - App description
 - Splash screen colors
- Service worker registration
- Cache management

App Manifest

— — —

- Tells the device about the app
- Add via link in html head with rel=manifest
- Icons
 - At least provide 512x512 and 192x192 icons
- Add to Home Criteria
 - <https://developers.google.com/web/fundamentals/app-install-banners/#criteria>

Service Workers

— — —

- Runs off the main thread
- Cannot access the window or DOM
- Can intercept fetch calls
- Can manage cache
- Can send/receive messages to/from main thread
- Workbox
 - <https://developers.google.com/web/tools/workbox/>
 - Built and maintained by Google
 - Helps accomplish common service worker tasks

Examples

— — —

- <https://github.com/jmcriffey/progressive-web-app>