Progressive Web App

Q:PWA

A demo of what the Q:CYBER suite would look like as a Progressive Web App

Scott C. Krause | 27 July 2021

Enter the PWA

The Why - Progressive Web Apps (PWA) are built and enhanced with modern APIs to deliver enhanced capabilities, reliability, and installability while reaching anyone, anywhere, on any device with a single codebase.

Enter the PWA

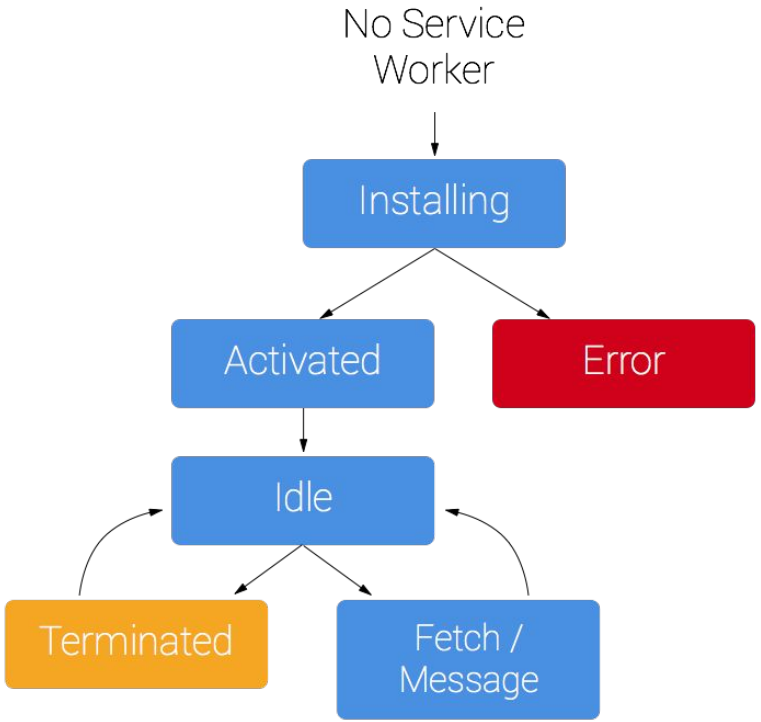
Expanding what the web can do with features like file system access, media controls, app badging, and full clipboard support. All of these capabilities are built with the web's secure, user-centric permission model, ensuring that going to a website is never a scary proposition for users.

Between modern APIs, WebAssembly, and new and upcoming APIs, web applications are more capable than ever, and those capabilities are only growing.

When a Progressive Web App moves out of a tab and into a standalone app window, it transforms how users think about it and interact with it.

Enter the Service Worker

- Separate Thread
- Acts as Network Proxy
- Returns a 200 status when fetched for cache
- No global state
- Does not communicate with DOM
- Requires HTTPS



Demo Q:CYBER PWA

Demo Q:CYBER PWA

- Install
- Branded Shell / No longer in Chrome
- Application Icon / Spotlight Search Icon
- Push Notifications (SSE)
- Execute on Startup
- Resources: CLICK HERE

The way forward

We can benefit now from utilizing Service Workers and Push Notifications.

- Advanced Caching to optimize performance and manage API version migration
- Push Notifications to alert an end user of critical issues without even opening the application
- Predictively precache Graph nodes
- File system Save, Save As, Drag, and Open JSON and CSV files
- Resources: CLICK HERE