Cheatography

Progressive Web Applications Cheat Sheet

by [deleted] via cheatography.com/2754/cs/12178/

Introduction

PWAs were originally proposed by Google back in 2015, and were described with this all-too-simple description: **“A Progressive Web App uses modern web capabilities to deliver an app-like user experience.”** However, there is a great deal more to a PWA than “modern web capabilities”. At its core, a PWA is all about delivering a better, faster user experience, regardless of what browser or device they are using.
there are several criteria that Google laid out to better determine if an app fits the PWA mold. To be considered a Progressive Web App, the app must be:

Credit: Frank J. Ohlhorst
http://sdtimes.com/web-app-model-becomes-progressive/

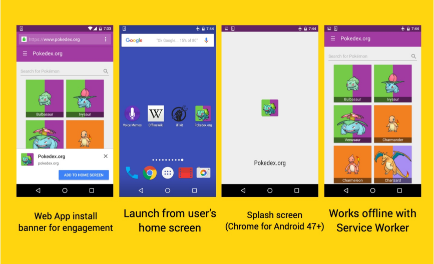
PWA Attributes

Progressive: In other words, the app should work for every user, regardless of browser choice. The app must be built with progressive enhancement as a core tenet.
Responsive: Adapt to any form factor, such as a tablet, mobile device, desktop, or any other device that consumes web content via a browser.
Connectivity independent: Ability to work offline or in poor connectivity situations.
App-like-Presentation: The app should use the app-shell model to provide app-style experiences.
Fresh: Always up-to-date, enabled by the service worker update process.
Safe: Delivered via HTTPS to protect data from interception, snooping or tampering.
Discoverable: Identified as applications, as defined by W3C manifests, and implementing a service worker registration scope, making them visible to search engines.
Re-engageable: Users are able to re-engage with applications through push notifications and other informational display features.
Installable: Allow users to have the freedom to install (or keep) apps on their device without having to rely on an app store.
Linkable: Can be shared via URLs, without the need for a complex installation process.

Under the Hood of the PWA Architecture

There are two parts to a PWA: service workers and application shell architecture. **Service workers** are bits of code that act like a proxy that sits between the website and the browser. Its job is to intercept what is asked of the browser and then hijack the responses given back. In effect, the service worker caches information, so the data is only fetched once for replay thousands of times. Service workers also exist to deliver extra features via browsers, which were impossible in the past. These include:. **Push notifications:** Informs the user of an event, such as a new message, a page being updated and so forth.
Synchronization: Updates data in the background, even when the user isn’t using the web page or website.
Caching: Stores information for offline use, allowing the user to have some functionality of a site while offline.
Pre-fetching data: Identifies data that may be used in the near future, such as images or content not yet displayed, and pre-fetches it to speed up access.
Incorporating additional data feeds: PWAs can query hardware such as geolocation, GPS, sensors, and so on using AJAX code.

PWA


Pokedex.org
ADD TO HOME SCREEN
Web App install banner for engagement
Launch from user's home screen
Splash screen (Chrome for Android 47+)
Works offline with Service Worker

PWA Advantages

No Installation Needed: PWAs are basically just web pages; users can consume them directly from their browser and then decide whether or not to keep them for offline use.
Instant Gratification: PWAs load instantly and do not require extensive prerequisites to run on the end user’s device.
Instant updates: PWAs are updated on use, just like a web page, which means PWAs will always be current with the latest deployed features and code.
Eliminates the App Store: Users do not need to access an app store to receive a PWA or add it to their desktop or favorites.
Portability: As websites, PWAs prove to be fully portable, and do not require any type of special packaging and deployment models.
Secure: PWAs are secure, hosted over HTTPS, using TLS between the endpoint and server.
Responsive: PWAs use responsive web design (RWD) techniques, which allows them to work on all combinations of mobile devices, laptops, smart phones and tablets.

PWA Disadvantages

Single sign-on and cross-application logins are not supported: Third-party applications that require a login, such as Facebook and Google, will continue to require their individual login, as PWAs are unable to independently collect and store that login information.
Lack of hardware functionality support: PWAs do not support all the hardware components that native apps support, such as cameras, GPS, and biometric accessories.
Browser compatibility: Newer versions of Chrome, Opera, and Samsung’s Android browser support PWA. However, IE, Edge, Safari and many custom and proprietary default browsers do not support PWA.