

A SEMINAR SYNOPSIS ON

Progressive Web Apps

Presented by
Aravind B 1ST14CS023

Progressive Web Apps

ABSTRACT

Progressive Web Apps (PWAs) are web applications that are regular web pages or websites, but can appear to the user like traditional applications or native mobile applications. The application type attempts to combine features offered by most modern browsers with the benefits of mobile experience. On the whole, Progressive Web Apps describe a collection of technologies, design concepts, and Web APIs that work in tandem to provide an app-like experience on the mobile web

Since around 2005 web development technologies have shifted from static to dynamic documents driven by server (PHP, ASP.NET) and client-side tools, and responsive web design. Despite an early push for web-based apps based on these technologies on devices such as the 2007 iPhone, attempts at web-apps failed by comparison to native-apps. Native apps provided a better user experience and booted faster compared to having to load in a browser at runtime. Packaged resources and direct access to hardware allowed native apps to perform much faster and to provide more features. By the mid-2010s, however, continued enhancements in HTML5, CSS3, and JavaScript, significantly more capable and standards compliant web browsers, along with powerful processors such as the A10 and Snapdragon 821 made performant hybrid-apps a viable alternative.

INTRODUCTION

A Progressive Web App is an app that can provide additional features based on what the device supports, providing offline capability, push notifications, an almost native app look and speed, and local caching of resources. This technique was originally introduced by Google in 2015, and proves to bring many advantages to both the developer and the users.

Developers have access to building almost-first-class applications using a web stack. This is always considerably easier and cheaper than building native applications, especially when considering the implications of building and maintaining cross-platform apps.

Developers can benefit from a reduced installation friction, and at a time when having an app in the store does not actually bring anything in terms of discoverability for 99,99% of the apps, Google search can provide the same benefits if not more.

A Progressive Web App is a website which is developed with certain technologies that make the mobile experience much more pleasant than a normal mobile-optimized website.

CHARACTERISTICS OF A PROGRESSIVE WEB APP (PWA)

It almost feels like working on a native app, as it offers the following characteristics:

- 1. Progressive**

Work for every user, regardless of browser choice

- 2. Responsive**

Fit any form factor: desktop, mobile, tablet, or forms yet to emerge

- 3. Connectivity independent**

Service workers allow work offline, or on low quality networks.

- 4. App-like**

Feel like an app to the user with app-style interactions and navigation.

- 5. Fresh and updated**

Always up-to-date thanks to the service worker update process.

- 6. Installable**

Allow users to “keep” apps they find most useful on their home screen without the hassle of an app store.

- 7. Linkable**

Easily shared via a URL and do not require complex installation.

HOW IS IT DIFFERENT FROM NATIVE MOBILE APPS

Native mobile apps are the most obvious way to build a mobile app. Objective-C or Swift on iOS, Java /Kotlin on Android and C# on Windows Phone. Each platform has its own UI and UX conventions, and the native widgets provide the experience that the user expects. They can be deployed and distributed through the platform App Store.

The main pain point with native apps is that cross-platform development requires learning, mastering and keeping up to date with many different methodologies and best practices

TECHNOLOGIES USED

Three main technologies are used in PWAs :

The Service Workers

A PWA must always work online as it is one of its main characteristics. Since the thing that allows the web app to work offline is the Service Worker, this implies that Service Workers are a mandatory part of a Progressive Web App.

A Service Worker is a JavaScript file that acts as a middleman between the web app and the network. Because of this it can provide cache services, speed the app rendering, and improve the user experience.

For security reasons, only HTTPS sites can make use of Service Workers, and this is part of the reason why a Progressive Web App must be served through HTTPS. Service Workers are not available on the device the first time the user visits the app. On the first visit the service worker is installed, and then on subsequent visits to separate pages of the site, this Service Worker will be called.

The App Manifest

The App Manifest is a JSON file that you can use to provide the device information about your Progressive Web App. You add a link to the manifest in every header on each page of your web site:

```
<link rel="manifest" href="/manifest.webmanifest">
```

This file will tell the device how to set:

- The name and short name of the app
- The icons' locations, in various sizes
- The starting URL, relative to the domain
- The default orientation
- The splash screen

The App Shell

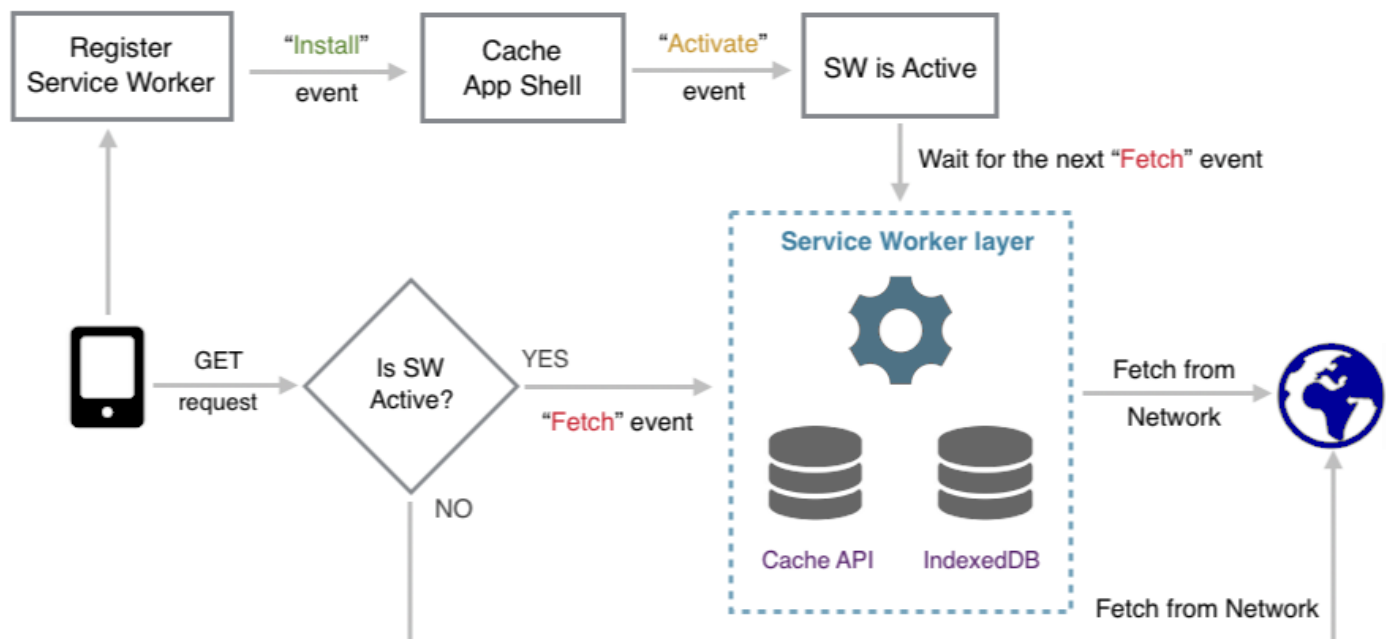
The App Shell is not a technology but rather a design concept. It's aimed at loading and rendering the web app container first, and the actual content shortly after, to give the user a nice app-like impression.

Some progressive web apps use an architectural approach called the App Shell Mode. For rapid loading, service workers store the Basic User Interface or "shell" of the responsive web design web application. This shell provides an initial static frame, a layout or architecture into which content can be loaded progressively as well as dynamically, allowing users to engage with the app despite varying degrees of web connectivity. Technically, the shell is a code bundle stored locally in the browser cache of the mobile device.

Caching

The App Shell is cached separately from the contents, and its setup so that retrieving the shell building blocks from the cache takes very little time.

ARCHITECTURE



ADVANTAGES OF PWA

1. PWA are lighter. Native Apps can weigh 200MB or more, while a PWA could be in the range of the KBs.
2. There's no native platform code, as the apps run on the browser, and not on the device.
3. The cost of acquisition is lower (it's much more difficult to convince a user to install an app than to visit a website to get the first-time experience).
4. Significantly less effort is needed to build and release updates.
5. They have much more support for deep links than regular app-store apps.

DISADVANTAGES OF PWA

1. Except Google Chrome, many browsers don't fully support PWA
2. Limited to Android devices. PWAs are not available on iOS as of now
3. Not available via Play Store. PWAs are installed by visiting a webpage and hence aren't available on app stores.
4. Maybe unsafe. Spam or fraudulent websites may harm your devices by installing their PWA.

CONCLUSION

Progressive Web Apps are the future of web development technologies and methods that allow us to perform our tasks on web browsers instead of our traditional native apps on mobile platforms.