MEET UP

PROGRESSIVE WEB APPS FOR FAST, RELIABLE AND ADAPTIVE WEBSITES

Progressive Web Apps (PWA's) aim to bridge the gap between native and web-based apps. During a meetup at Bi4 Group, the Engineering Team discussed the technology behind PWA's, how to build them and best use cases.

WHAT ARE PWA'S?

Progressive Web Apps (PWA) are apps that sit between native and web-based apps, offering the best of both: a reliable, fast and adaptive native web experience without the hassle of installing an app through an app store. Furthermore, PWA's are platform-, client and user-independent, which means that a developer only needs to build one single app that can be used anywhere by anyone.

A typical PWA consists of three basic building blocks: a service worker, an application shell and an app manifest. Service workers are scripts written in JavaScript that act like a client-side proxy that put the app in control of the cache, allowing error-free app display during offline access, and respond to source requests. An application shell provides an initial static frame on which content can be loaded progressively and dynamically, so that users can engage with the app, whether they're connected to the internet or not. Finally, an app manifest is a json file containing metadata associated with the web application. It was introduced by Google, that uses it to index PWA through their search engine and verify if it's dealing with an authentic PWA.

How to install and use a PWA?

PWA's can be installed and used easily. If a website offers a PWA, it can be added to the user's home screen after visiting a website through a browser. What is installed next is a slim, skeletal app that is enriched with data once users start to use the app more. Through an app icon, the app can be run from the home screen. The app can also be run offline, using the cached data from an earlier online site visit.

How to develop a PWA?

Because PWA's adapt to a underlying programming paradigm, all a developer needs to know in order to be able to build PWA's is HTML, CSS and JavaScript. PWA's adapt to all languages and frameworks currently used at Bi4 Group, such as Python, Ruby, .NET, ReactJS, AngularJS and Angular. There are now many resources available online for developers so they don't have to build everything themselves. For example, Google developed and open-sourced Lighthouse, an automatic tool to improve the quality of a PWA, that eliminates much of the previously required manual testing.

WHY ARE PWA'S SO POPULAR?

Today's mobile users spend more time on native apps than on the web, because they offer a smooth experience. However, this requires them to visit an app store, download the native app, agree with its terms of use and update when necessary – a lot of hurdles that native mobile app users aren't willing to take these days. PWA's offer a quicker way of downloading and using apps compared to native apps. Google, Microsoft and Apple see PWA's as the next step in the evolution of apps and now offer more PWA support for both app developers and end users. Speed is another reason why PWA's are popular: evidence shows that if an app takes more than three seconds to load, it will lose 53% of its user base immediately. PWA's respond to the need for fast load times, in combination with a reliable, engaging experience so that a user is likely to return to your website and spend more time online, which is good from an advertising perspective as well.

DRAWBACKS OF PWA'S

Even though some think that PWA's will replace native apps, there are currently still a number of drawbacks for users and developers. For example, because there's no app store to search for a specific PWA, people who search for your app there won't be able find it. This means you're missing out on important traffic. Also, not all browsers provide full support for PWA's, or worse, some PWA's will only work with a certain browser and version number, such as Chrome.

Required logins on the web is another drawback, as both Google and Facebook cannot fetch data from their apps. PWA's cannot access all hardware of a smartphone like native apps do, such as your camera. Finally, PWA's may not be the best option for your website, as they offer limited performance for computational-heavy operations due their slim size. The ideal PWA use case therefore is a static website: one that not changes over time will offer better offline performance.

BRIGHT FUTURE

At the moment, the future looks bright for PWA's: the technology is becoming more mature every day. More users enjoy the fast, reliable and adaptive experience offered by PWA's. Also, more companies are turning to PWA development. One single app that can be used on different platforms means less development time and costs for them. Software companies such as Google, Microsoft and Apple are offering more support for PWA's as well through their app stores and browsers.