

Progressive Web Apps a New Way to Experience Mobile

Bharath Kumar NJ,
 3rd year CSE,
 AMC Engineering College,
 18th km Bannerghatta Main Road, Bangalore – 560083

Abstract—The latest trendy technology seems to be Progressive web apps, which the greater web community – most notably Google has been pushing heavily. These progressive web apps make the whole website looks like a native apps with offline working property.

Keywords— Web Apps, Manifest, Website, Service Worker

I. INTRODUCTION

The world is on the internet today, about 46.1% of the world’s population uses the internet, and people are trying to pull in the best use of Technology to attain the more quality in work. Whatever the last knot of any technology is a mobile phone and we all knew how much impact the smartphone have created today, operating system like: Android, iOS are ruling the market hence the craze of Application’s is reaching sky. As per the statistics the Google Play Store have 2 million apps, Apple app store have 1.5 million apps, and day by day the number of apps are scaling rapidly. But how many apps one can install in his/her phone? Not more than the capacity of phone memory or external devices, we feel to have more and more application which can help us to access things in one click, but we couldn’t able to accomplish our wish with these native apps. The new technology rolling out is “Progressive Web Apps”, hope this will ease our wish.

II. WHAT ARE PROGRESSIVE WEB APPS?

Progressive Web Apps are experiences that combine the best of the web and the best of apps. They are useful to users from the very first visit in a browser tab, no install required. As the user progressively builds a relationship with the App over time, it becomes more and more powerful. It loads quickly, even on flaky networks, sends relevant push notifications, has an icon on the home screen and loads as top-level, full screen experience.

There have been a few turning points in the history of the web platform that radically changed how web apps were built, deployed & experienced. Ajax was one such pivot that led to a profound shift in web engineering. It allowed web applications to be responsive enough to challenge the conventional desktop apps. However on mobile, the experience was defined by native apps and web apps hardly came close to them, at least until now.

III. PROGRESSIVE WEB APPS ARE

- Progressive - Work for every user, regardless of browser choice because they’re built with progressive enhancement as a core tenet.

- Responsive - Fit any form factor: desktop, mobile, tablet, or whatever is next.
- Connectivity independent - Enhanced with service workers to work offline or on low quality networks.
- App-like - Feel like an app to the user with app-style interactions and navigation because it's built on the app shell model.
- Fresh - Always up-to-date thanks to the service worker update process.
- Safe - Served via HTTPS to prevent snooping and ensure content hasn’t been tampered with.
- Discoverable - Are identifiable as “applications” thanks to W3C manifests and service worker registration scope allowing search engines to find them.
- Re-engageable - Make re-engagement easy through features like push notifications.
- Installable - Allow users to “keep” apps they find most useful on their home screen without the hassle of an app store.
- Linkable - it's the web! [1]

IV. WHAT YOU WILL NEED

- A. *App Shell*: The app's shell, is the minimal HTML, CSS, and JavaScript that is required to power the user interface of a progressive web app and is one of the components that ensures reliably good performance. Its first load should be extremely quick, then immediately be cached. This means that the shell does not need to be loaded every time, but instead just gets the necessary content.

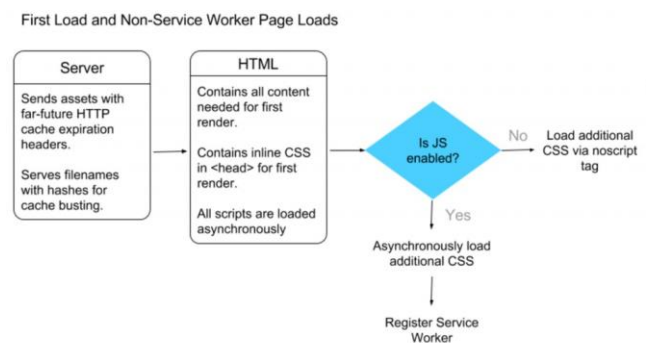


Fig. 4.1 Instant Loading Web Apps with an Application Shell Architecture.

B. *Service Worker*: The first step to making the app work offline is to register a service worker, a script that allows background functionality without the need for an open web page or user interaction.

```
<script>
if('serviceWorker' in navigator) {
  navigator.serviceWorker
  .register('/sw.js')
  .then(function() { console.log("Service Worker
  Registered"); });
}
</script>
```

Fig. 4.2 code to test whether browser supports service worker or not.

C. *Manifest*: Every app needs a manifest—a JSON-formatted file named manifest.json that describes the app, also it gives the ability to control how your app appears to the user in the areas that they would expect to see apps, direct what the user can launch and, more importantly, how they can launch it. In the future the manifest will give you even more control over your app.

```
{
  "short_name": "App Name",
  "name": "Full App Name",
  "icons": [
    {
      "src": "launcher-icon-2x.png",
      "sizes": "96x96",
      "type": "image/png"
    },
    {
      "src": "launcher-icon-3x.png",
      "sizes": "144x144",
      "type": "image/png"
    },
    {
      "src": "launcher-icon-4x.png",
      "sizes": "192x192",
      "type": "image/png"
    }
  ],
  "start_url": "/index.html",
  "display": "standalone",
  "orientation": "portrait"
}
```

Fig. 4.3 Example of Manifest Code.

The last thing we have to do is telling the browser about the manifest, once you have the manifest created and it is hosted on your site, all you need to do is add a link tag from all your pages that encompass your app, as follows

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

Fig. 4.4 Linking Manifest to your App. [4]

This much will make a website that looks like an app, just by adding icon to home screen of your mobile phone. Hope we have not installed anything from store which will not consume so much of space in our device than a KB of icon, these web apps allows us to work offline with the help of service worker, and updated dynamically when device is connected to internet, although HTML5 have cache API, but it will be deprecated in coming days.

Here is a Simple Progressive Web App which I have developed, link: tinyurl.com/litev2 (open in mobile browsers to experience the app)

ACKNOWLEDGMENT

All the Authors are Thankful to this Technology and Special thanks to the Mentor Shruti Balasa ma'am and faculties for their support & guidance for this paper presentation.

REFERENCES

- [1] Alex Russel, (Creator of JavaScript) "Article on progressive Web Apps".
- [2] Google Developers Article on Progressive Web Apps.
- [3] Google Developers Article on Progressive Web Apps.
- [4] Google Developers Article on Progressive Web Apps.