Pwa execution tools

January 10, 2024.

Contents

1 Introduction

In this document will be explained the pwa applications and the characteristics for the browser that support the pwa or technical requirements for the pwa, the compatibility of the pwa with the browser and the operating system, and the tools that we are going to use to develop the PWA.

1.1 Purpose

The purpose of exploring the topic "PWA execution Tools" is to identify, define, and evaluate the various tools essential for the effective execution and deployment of Progressive Web Apps (PWAs).

1.2 Definitions, Acronyms, and Abbreviations

In this document, the following terms, acronyms, and abbreviations are utilized:

- PWA: Progressive Web Application.Web applications employing modern web capabilities to offer a native-like user experience.
- Web: A client-server system, where the client is a web browser and the server is a web server.
- Polymer: An open-source JavaScript library for building web applications using web components.
- Web Manifest: A JSON file providing metadata to the browser about the application, enabling features like 'Add to Homescreen.'
- HTTPS: HyperText Transfer Protocol Secure. A protocol ensuring secure communication between the user's browser and the web server.
- Service Worker: A script running in the background, facilitating event-driven functionalities and offline capabilities for PWAs.
- Application Shell: A foundational structure of a PWA, providing essential UI elements using HTML and CSS, enhancing user experience.
- Compatibility: The ability of a PWA to function optimally across various browsers and operating systems.

1.3 References

References

[1] Anh, D. T. (2023, August 30). Top Best PWA development Tools and Technologies for your business. Magenest - One-Stop Digital Transformation Solution. `https://magenest.com/en/pwa-development-tools/`

[2] MDN Web Docs. (2023, November 17). How to make PWAs installable - Progressive Web Apps. Retrieved January 9, 2024, from `https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Tutorials/js13kGames/Installable_PWAs`

[3] Herrera, I., Herrera, I. (2023, July 27). Compatibilidades de las PWA según el navegador y el sistema operativo. ttandem.com. Retrieved January 9, 2024, from https://www.ttandem.com/blog/desarrollo-que-son-las-pwa-o-progressive-web-applications/compatibilidades-de-las-pwa-segun-el-navegador-y-el-sistema-operativo/

1.4 Overview of the Document

This document delves into the crucial aspect of tools used for the execution of Progressive Web Apps (PWAs). The primary focus is on identifying and defining the requirements and browsers essential for PWA execution, as well as exploring the specific tools designed to facilitate PWA development and deployment.

2 General Description of the Document

Pwa (progressive web application) is a type of application that is developed with web technologies, but that has the characteristics of a native application, such as the ability to work offline, send notifications, access the camera, etc. In addition, they can be installed on the device and are updated automatically. They are developed with HTML, CSS and JavaScript, and can be used on any device, regardless of the operating system. They are also responsive, so they adapt to any screen size. They are also indexed by search engines, so they can be found in the same way as a website. They are also progressive, so they work on any browser, regardless of the version. They are also safe, since they are served through HTTPS, so they cannot be intercepted. They are also linkable, so they can be shared through a URL. They are also updated automatically, so the user always has the latest version. They are also installable, so they can be installed on the device's home screen. They are also independent of the connectivity, so they can work offline. They are also re-engageable, so they can send notifications to the user.

- Installation: Supporting browsers can prompt users to install the PWA, functioning similarly to native apps.
- Compatibility: PWAs adapt to various browsers and operating systems, offering optimal user experiences.
- Device-Specific Functionality: The functionalities of a PWA may vary depending on device and browser compatibility.

2.1 Pwa Requirements

A Progressive Web Application (PWA) requires essential technical components or requirements for its functionality:

- Web Manifest File: A web manifest is a JSON file that provides metadata about a PWA. It includes information such as the app's name, icons, theme colors, and display modes, and is importan because The web manifest enables the browser to understand how to display the PWA when installed on a user's device. It contributes to a more app-like experience, allowing users to add the PWA to their home screen.

- HTTPS Security: PWAs require a secure connection using HTTPS (Hypertext Transfer Protocol Secure). This ensures that data exchanged between the user and the PWA is encrypted and secure. HTTPS is essential for protecting user data and ensuring the integrity and security of the PWA. It also allows the PWA to leverage certain modern web features, such as service workers, which are crucial for offline capabilities

- Service Worker Integration: A service worker is a script that runs in the background, separate from the web page, enabling features like offline support, push notifications, and background synchronization. Service workers play a key role in enhancing the performance and functionality of PWAs. They enable the caching of resources, allowing the PWA to work offline or in low-connectivity situations, improving user experience.

- Application Shell or Common HTML/CSS: The application shell is the minimal HTML, CSS, and JavaScript required to load the PWA's user interface. It provides a basic structure that is cached, allowing for quicker loading times on subsequent visits. is important becasue mplementing an application shell is part of the PWA's strategy for fast initial loading and smooth navigation. This technique helps create a responsive and engaging user experience by delivering essential UI components quickly.

2.2 Pwa browser

The browser prompts the user to install the PWA if it supports the installation process.

- Chrome: Chrome supports the installation of PWAs on Android, Windows, and Linux. becasue is the most used browser in the world and is the most compatible with PWA.

- Safari: Safari supports the installation of PWAs on iOS and macOS becasue Safari is the default browser of Apple devices.

- Firefox: Firefox supports the installation of PWAs on Android, Windows, and Linux becasue is the second most used browser in the world and it have a good compatibility with PWA.

2.3 Pwa Tools

the tools that we are going to use are:

- Polymer: is a JavaScript library for building web applications using web components. It is developed by Google and enables the creation of reusable components that can be easily integrated into web applications.

- Magento PWA Studio: is a set of tools for building Progressive Web Apps (PWAs) on top of the Magento platform. It provides a development environment, tools, and libraries to create high-quality, performant PWAs for Magento-based e-commerce stores.

- Ionic: is a popular open-source framework for building cross-platform mobile applications using web technologies such as HTML, CSS, and JavaScript/TypeScript. It comes with a rich set of UI components and can be used to build both web and mobile apps.

- React PWA Library: This could refer to various libraries and tools built around React for developing Progressive Web Apps. React itself is a JavaScript library for building user interfaces, and several libraries and frameworks exist to enhance the development of PWAs using React, such as Create React App for bootstrapping projects.

- Vue Storefront: Storefront is a standalone PWA storefront for eCommerce. It is built with Vue.js and designed to work with various eCommerce backends, providing a fast and engaging shopping experience for users.

- Angular PWA: refers to the progressive web app capabilities provided by the Angular framework. Angular is a TypeScript-based web application framework developed by Google, and it includes features and tools for building PWAs.

3 Appendices

- Progressive Web App (PWA): A type of application software delivered through the web, built using common web technologies and designed to work on any platform with a standards-compliant browser.
- Web Manifest: A JSON file containing metadata about a PWA, providing details such as the app's name, icons, theme colors, and display modes.