

Cache in Pwa, Is important? and what is it strategies

March 15, 2024.

Contents

| | | |
|----------|---------------------------|----------|
| 1 | Introduction | 3 |
| 1.1 | Purpose | 3 |
| 1.2 | References | 3 |
| 2 | Cache | 3 |
| 2.1 | functionabiliry | 3 |
| 2.2 | Strategies | 3 |

1 Introduction

In this document we will talk about the importance of the cache in the PWA, and how the service worker is used to cache resources and provide offline support.

1.1 Purpose

The purpose of this document is to provide a detailed description of the cache and how to use it in a PWA and its strategy to improve the performance of the web application.

1.2 References

References

- [1] Service Worker API - Web APIs (2024, 8 febrero) MDN Web Docs. February 29, 2024, from https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API
- [2] Educative. (2022, March 18). Educative Answers - Trusted Answers to Developer Questions. February 29, 2024, from <https://www.educative.io/answers/what-are-the-pros-and-cons-of-using-service-workers-in-pwas>

2 Cache

The cache is a feature of a PWA that is the ability to explicitly cache some of the app's resources so that they can be used offline. This is done using the service worker, which is a script that runs in the background of the web application, separate from the main thread. The service worker can intercept network requests and cache the responses, allowing the web application to work offline or in low-quality networks.

2.1 functionality

It can be used for offline operation, its caching enables in a PWA to function to a greater or lesser extent while the device does not have an internet connection. This is done by caching the resources that the web application needs to work, such as HTML, CSS, JavaScript, and images, and serving them from the cache when the network is not available.

2.2 Strategies

- **cache Online:** It returns a resource from the cache without making a network request, if the resource is not in the cache, it will make a network request to fetch it. This strategy works for serving assets pre-cached for the installation of a service.
- **network only :** It needs a network to get a resource. Good if stale or cached version is unacceptable.
- **cache first:** It returns a resource from the cache without making a network request, if the resource is not in the cache, it will make a network request to fetch it. This strategy works for serving assets pre-cached for the installation of a service.

-
- network first: It return a resource from the network without making a cache request, if the resource is not in the network, it will make a cache request to fetch it. this strategy works for servicing assets pre-cached for the installation of a service.
 - Stale while revalidate: It return a resource from the cache without making a network request, if the resource is not in the cache, it will make a network request to fetch it. this strategy works for servicing assets pre-cached for the installation of a service.