



UTT

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

GOBIERNO DE BAJA CALIFORNIA

Cache with network and offline mode in PWA

Name:	López Bautista Cristian Alexis
Group:	10-B
Subject:	Progressive Web Applications
Professor:	Dr. Ray Brunet Parra Galaviz
Date:	Wednesday, March 20th, 2024

1 Cache with Network

Cache with Network is a strategy that involves using both caching and network resources to deliver content. This strategy can be implemented in several ways, including:

1.1 Cache First, Network Fallback

Cache first, Network fallback.

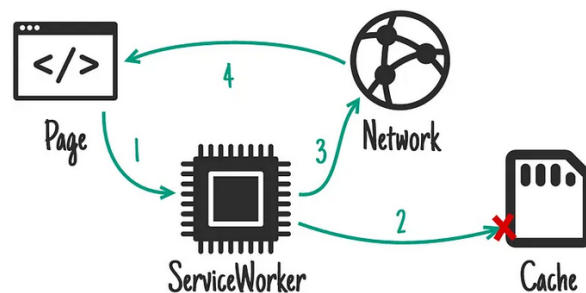


Figure 1: Cache first diagram.

This strategy is also known as “Cache falling back to network”. In this strategy, the service worker first tries to fetch the resource from the cache. If the resource is in the cache, it’s returned directly to the user. If the resource is not in the cache, the service worker fetches it from the network, delivers it to the user, and adds it to the cache for future use. This strategy is useful for resources that don’t change often, like CSS or JavaScript files.

1.2 Network First, Cache Fallback

The network first, Cache fallback.

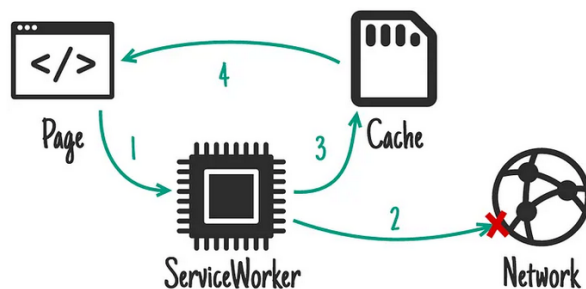


Figure 2: Network first diagram.

This strategy is also known as “Network falling back to cache”. In this strategy, the service worker first tries to fetch the resource from the network. If the network request is successful, the response is delivered to the user and also updated in the cache. If the network request fails (for example, because the user is offline), the service worker fetches the resource from the cache and delivers it to the user. This strategy is useful for resources that change frequently or need to be up-to-date, like a news article or a social media feed.

1.3 Stale While Revalidate

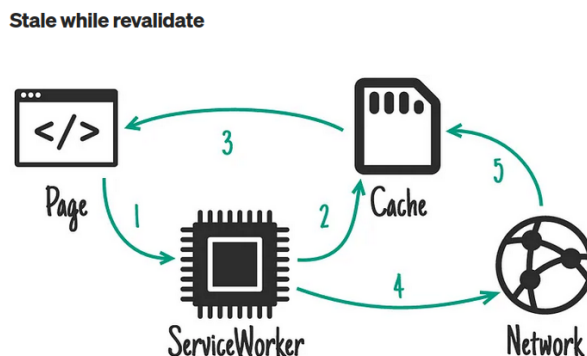


Figure 3: Network first diagram.

In this strategy, the service worker first delivers the resource from the cache (if available) to the user, making for a fast response. In the background, the service worker fetches the resource from the network and updates the cache. This means that the user might get stale data on the first load, but subsequent loads will have the most up-to-date resource. This strategy is useful for resources where speed is more important than freshness, like static assets or an app shell.

Each of these strategies has its own strengths and weaknesses, and the choice of strategy depends on the specific needs of your application and the nature of the resources being requested. For example, for resources that need to be up-to-date, a network-first strategy might be more appropriate, while for resources that don't change often, a cache-first strategy could be a better choice. It's also possible to use different strategies for different types of resources in the same application. For example, you might use a cache-first strategy for your CSS and JavaScript files, and a network-first strategy for your news articles.

2 Offline mode

The offline mode in Progressive Web Applications (PWAs) is a key feature that allows the app to still function even when the user's device is not connected to the internet. This is made possible by technologies such as Service Workers and the Cache API.

2.1 Service Workers

These are scripts that your browser runs in the background, separate from a web page, opening the door to features that don't need a web page or user interaction. They can intercept network requests and manage responses, which is crucial for handling offline scenarios.

2.2 Caching

Service Workers can cache important files (like HTML, CSS, JavaScript, and images) and serve them directly from the cache when the user is offline. This means that even without an internet connection, users can still load a basic version of the app.

2.3 Background Sync

This feature allows a web app to defer actions until the user has stable connectivity. This is especially useful for ensuring that whatever the user wants to send is actually sent.

2.4 Offline-first Architecture

In this approach, the design of the app starts with offline functionality as a core feature. The app is built with the assumption that it might not have network connectivity, and any data that needs to be updated is done so when the connection is available.

2.5 App Shell Architecture

This is a design concept where the initial load of a mobile web app provides a basic shell of an app's UI (like the header, footer, and navigation), and the content for the app is loaded after. This can ensure that users can get meaningful pixels on the screen without the network, making the app feel instant and reliably fast.

3 Conclusion

In conclusion, service workers play a pivotal role in shaping the future of web development by bridging the gap between web and native applications. With their ability to provide offline functionality, efficient caching, and background synchronization, service workers empower developers to create web applications that rival the performance and user experience of their native counterparts. By embracing service workers, developers can deliver fast, reliable, and engaging web experiences that adapt seamlessly to varying network conditions. As the web continues to evolve, service workers will remain a cornerstone technology, driving innovation and pushing the boundaries of what's possible on the modern web.

4 Bibliography

1. Babu, L. V. (2021, 27 december). Best Caching strategies — Progressive Web App (PWA). Medium.
<https://medium.com/animall-engineering/best-caching-strategies-progressive-web-app-pwa-c610d65b2009>
2. Caching - Progressive web apps — MDN. (2023, 25 october). MDN Web Docs.
https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Guides/Caching
3. Offline and background operation - Progressive web apps — MDN. (2023, 10 december). MDN Web Docs.
https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Guides/Offline_and_background_operation