

Progressive Web Apps



Was sind 'Progressive Web Apps'?

- Grundsätzlich Websites mit:
 - einer App-Like User Experience
 - einem Client-Side-Proxy (ServiceWorker)
 - Funktionen welche normalerweise nativen Anwendungen vorbehalten sind, z.B:
 - Push-Notifications/Benachrichtigungen
 - Offline Verfügbarkeit
 - Home-Screen-Präsenz
 - ...

Warum sollte man 'Progressive Web Apps' verwenden?

- Ein Projekt für alle Plattformen
- Bessere Erreichbarkeit der Benutzer da:
 - Die Reichweite von Websites kombiniert wird mit Vorteilen/Funktionen welche sonst nur Apps vorweisen können.
 - Der reibungslose Zugang mittels URL (Vorteil aller Websites)
 - (höhere) Netzwerk-Unabhängigkeit
 - Push-Notifications
 - (leichteres Speichern) Spezifischer Benutzerinformationen
 - Home-Screen-Präsenz

Wie funktionieren 'Progressive Web Apps'?

Manifest.json

- Informiert den Browser, dass es sich um eine PWA handelt (Alle grundlegenden Browser unterstützen PWA's: Chrome, Firefox, Edge, Safari,...)
- Gibt grundsätzliche Metadaten wie Name oder App-Icon an.

Service Workers

- Ist ein Client-Side Proxy
- Gibt der Progressive Web App ihre einzigartigen Funktionalitäten
- Läuft in einem eigenen Prozess
- Gibt Kontrolle über die Netzwerkanfragen (fängt die Netzwerkanfragen ab und ermöglicht die Daten aus dem Cache zurückzugeben)
- Ermöglicht es das Cache-Verhalten der Website zu definieren
- Ermöglicht Push-Messages
- Baut auf der Fetch-API auf

Service Worker - Fetch

- Library um Netzwerkanfragen durchzuführen
- Ersetzt XMLHttpRequest
- Das Fetch-Event ist meist der Einstieg für den Service Worker (hier werden Netzwerkanfragen abgefangen)
- Promise-Based

Promise API

- Für Asynchrone Programmierung
- `fetch(...)`
 - `.then(response => { return response.json() })`
 - `.then(json => {...})`
 - `.catch(err => { })`

Service Workers - Lifecycle

No Service
Worker



Installing

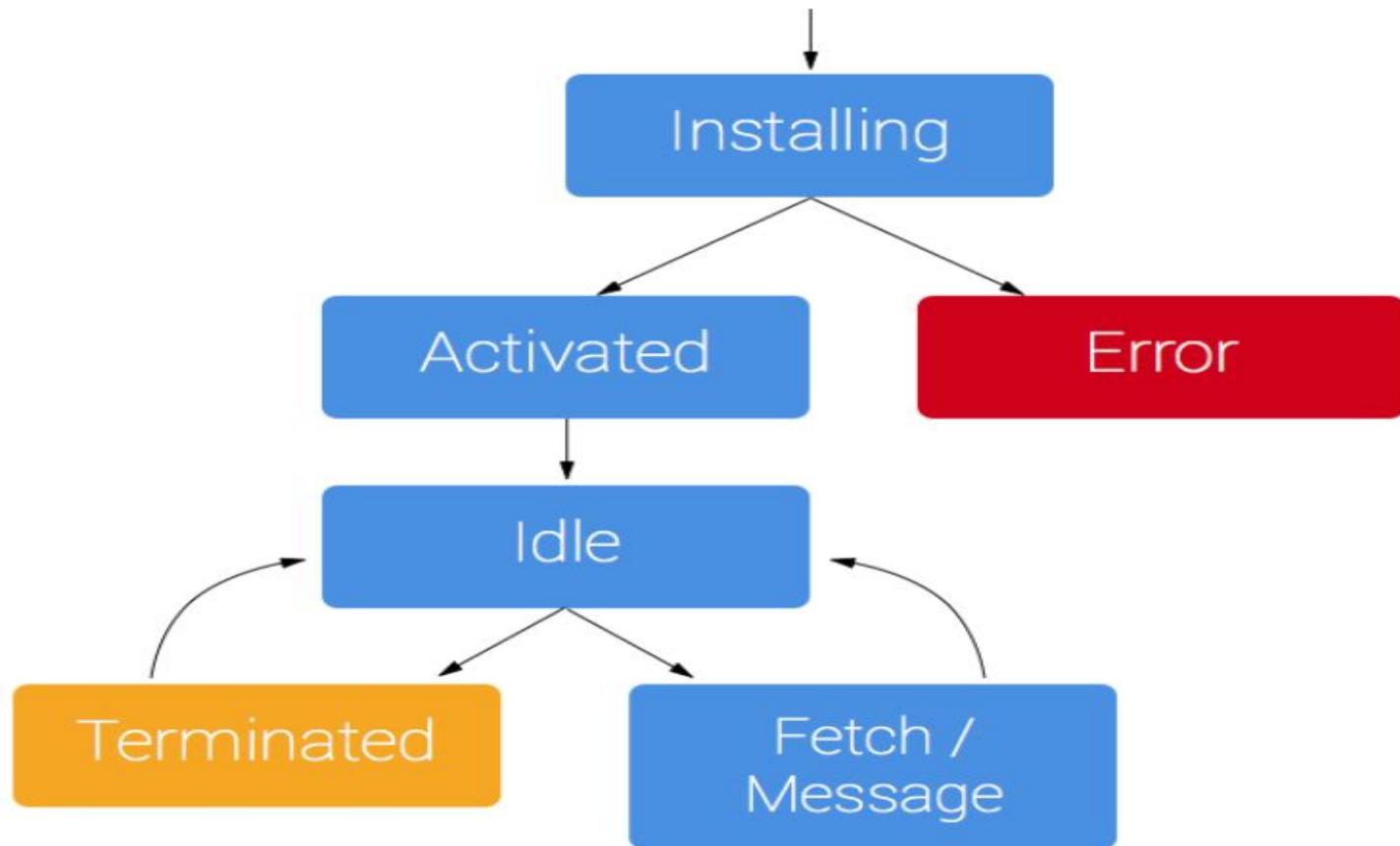
Activated

Error

Idle

Terminated

Fetch /
Message

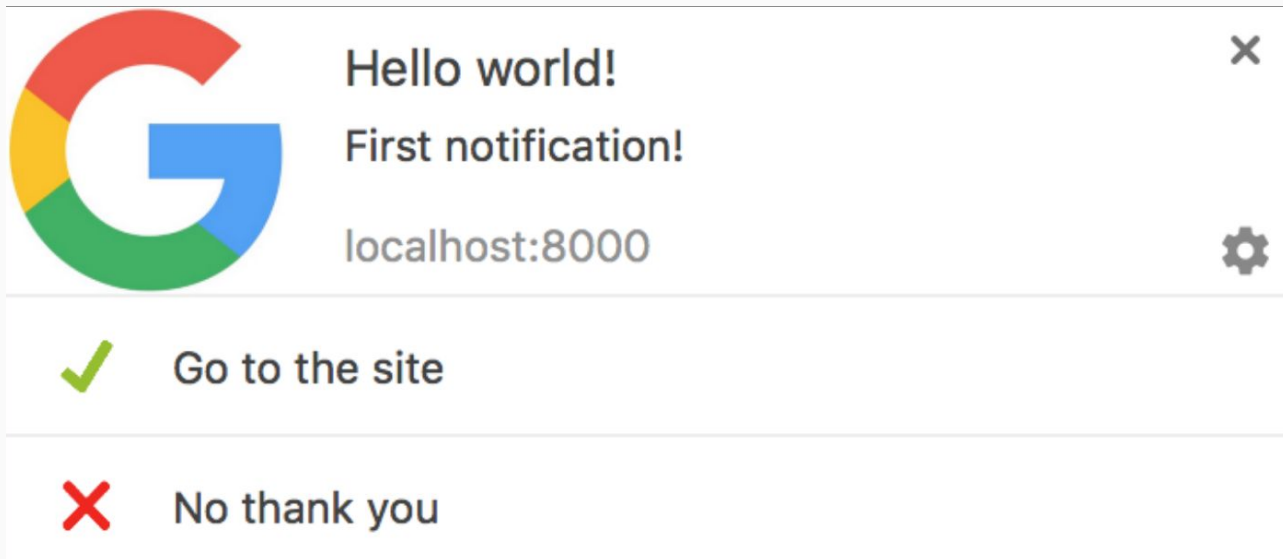


Service Workers - Events

- **install**
 - Wird aufgerufen wenn der Service Worker erfolgreich installiert wurde (In diesem Zustand kann ein anderer Service Worker noch arbeiten!)
- **activate**
 - Wird aufgerufen wenn der alte Service Worker beendet wurde und der neue Service Worker aktiv wird
- **fetch**
 - Wenn die Website Netzwerkanfragen ausführt werden diese zuerst an das Fetch Event des Service Workers geschickt, welcher die Anfragen an das Netzwerk weiterleiten kann
- ...

Service Workers - Notifications

- Notification API



Service Workers - Push Messages

- Client registriert sich in dem Push-Service des Browsers
- Ermöglicht es den Server Nachrichten an spezifische Clients zu senden
- Der Server sendet hierfür eine HTTPS-Anfrage an eine von dem Client generierte URL (mit einem vom Client-Generierten Public-Key)

Service Workers - Push Messages - Lifecycle



Quellen

- <https://developers.google.com/web/ilt/pwa/>
- <https://developers.google.com/web/fundamentals/instant-and-offline/offline-cookbook>