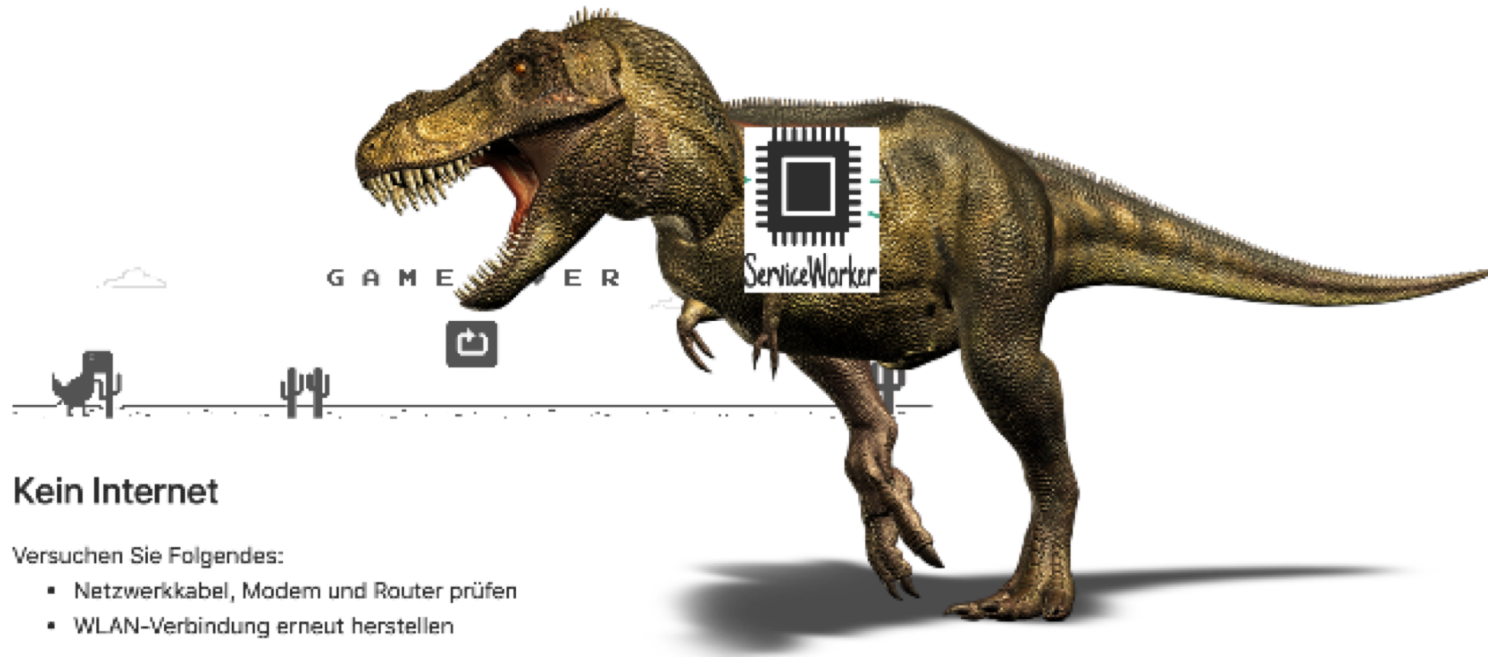


WORKSHOP

SERVICE WORKER



Kein Internet

Versuchen Sie Folgendes:

- Netzwerkkabel, Modem und Router prüfen
- WLAN-Verbindung erneut herstellen

ERR_INTERNET_DISCONNECTED

INHALT

01

WAS SIND SERVICE WORKER?

02

SERVICE WORKER LIFECYCLE

03

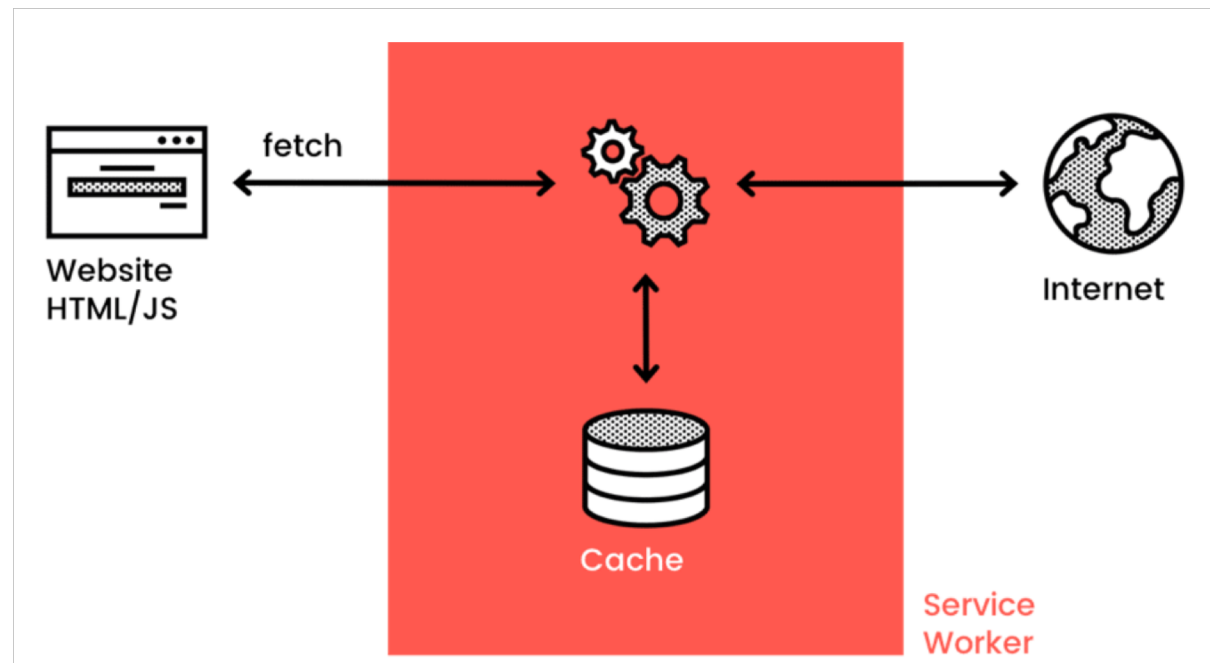
CACHING-STRATEGIEN

04

BACKGROUND SYNC

WAS SIND SERVICE WORKER?

- Skript, dass separat vom Browser läuft
- Service Worker befinden sich zwischen dem Webbrowser und dem Netzwerk
- Fungieren als Vermittler
- Ressourcen anfragen / Ressourcen dem Cache hinzufügen / Ressourcen aus dem Cache holen
- Offline-Verfügbarkeit der Web App (je nach Caching Strategie)

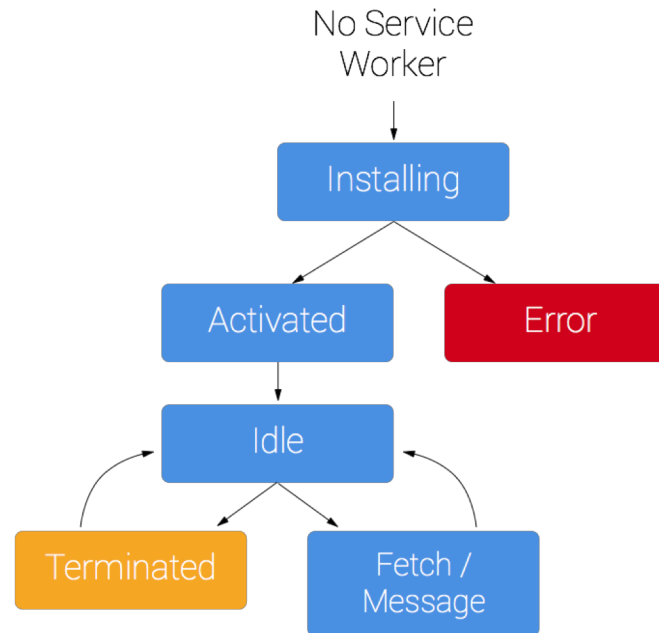


SERVICE WORKER

- Service Worker verfügt über ein **Cache Interface**, das den Entwicklern die Kontrolle gibt, welche Ressourcen/Daten gespeichert werden sollen
- **Entwickler entscheiden** über statischen und dynamischen Cache
- Entwickler kann selbst über die **Caching-Strategie** entscheiden

WAS SIND SERVICE WORKER?

SERVICE WORKER LIFECYCLE



- **Install:** Um den Service Worker installieren zu können muss dieser registriert werden
- **Activate:** Service Worker wird aktiviert (gute Gelegenheit um veraltete Caches zu löschen)
- **Idle:** Der Service Worker befindet sich im Leerlauf bis neue Events aufgerufen werden
- **Fetch:** Die vom Service Worker kontrollierten Ressourcen werden geladen
- **Terminated:** Der Service Worker wird nicht weiter benötigt und wurde beendet

SERVICE WORKER LIFECYCLE

- Anwendungsbeispiel (News PWA):

The screenshot displays a PWA News application interface. At the top, there is a blue header with the text "PWA NEWS" on the left and a hamburger menu icon on the right. Below the header, a filter dropdown menu shows "T3n" with a downward arrow. The main content area is a grid of ten news article cards, arranged in two rows of five. Each card features a representative image at the top, followed by a headline, a short introductory paragraph, and a blue "READ MORE" button at the bottom. The articles cover various topics including Bitcoin, IKEA, Stadtwerke München, Raspberry Pi, Beats headphones, Facebook, Native Advertising, Elektromobilität, and Nextory.

BITCOIN MIT SATTEM PLUS: 3 GRÜNDE FÜR DEN PLÖTZLICHEN KURSANSTIEG
Nachdem Kryptowährungen monatelang gebeutelt wurden, gab der Bitcoin kürzlich ein sehr deutliches Lebenszeichen von sich. Der Sprung um über 1.000 US-Dollar binnen einer Stunde kam aber nicht völlig unerwartet.

IKEA WILL MÖBEL BALD NICHT MEHR NUR VERKAUFEN
Kaufst du noch oder mietest du schon? Der Möbelriese Ikea experimentiert aktuell mit Leasing- und Mietmodellen für seine Möbel. Doch das dürfte sich nur für wenige Kunden lohnen.

STADTWERKE MÜNCHEN – WO INSTANDHALUNGSTEAMS IHRE EIGENE IOS-APP BEKOMMEN
Anzeige: Die Stadtwerke München bieten IT-Profis spannende Projekte und vielseitige Karrierechancen. Du kannst dir nicht genau vorstellen, was für Projekte das sind? Hier erfährst du es!

RASPBERRY PI: OFFIZIELLE MAUS UND TASTATUR VORGESTELLT
Die Raspberry Pi Foundation hat eine offizielle Maus und Tastatur für den Bastelrechner Raspberry Pi ins Verkaufsprogramm genommen. Beide Geräte sind recht günstig, es gibt sie in einem Himbeer-Farbtönen sowie in Grau und Schwarz.

BEATS POWERBEATS PRO: APPLE-TOCHTER KÜNDIGT TRUE-WIRELESS-KOPFHÖRER MIT „HEY SIRI“ AN
Apple-Tochter Beats hat mit den Powerbeats Pro neue In-Ear-Ohrstöpsel angekündigt, die mit dem gleichen H1-Chip wie die neuen AirPods versehen sind.

FACEBOOK: MILLIONEN NUTZERDATEN LANDEN IN DER AMAZON CLOUD

NATIVE ADVERTISING: DA LÄUFT WAS GEWALTIG SCHIEF

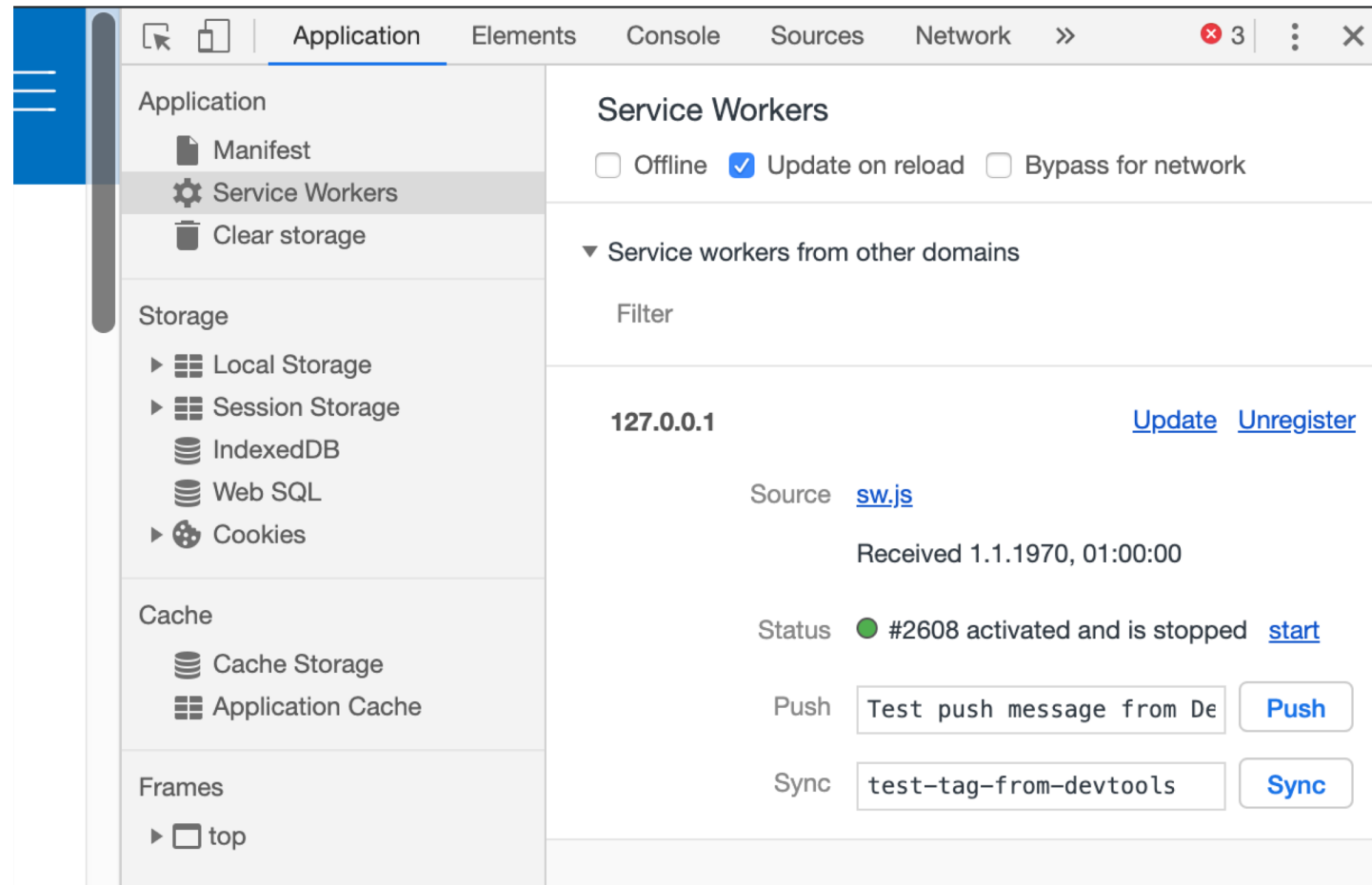
ELEKTROMOBILITÄT: POLNISCHES STARTUP WILL 200.000 ELEKTROAUTOS

SO VERBESSERN INTERNETNUTZER DIE ERDBEENÜBERWACHUNG

NEXTORY: NEUE E-BOOK- UND HÖRBUCH-FLATRATE STARTET IN DEUTSCHLAND

SERVICE WORKER LIFECYCLE

- Debugging



SERVICE WORKER

SERVICE WORKER LIFECYCLE

- Register:

```
if ('serviceWorker' in navigator) {
  window.addEventListener('load', function() {
    navigator.serviceWorker.register('/sw.js').then(function(registration) {
      // Registration was successful
      console.log('ServiceWorker registration successful with scope: ', registration.scope);
    }, function(err) {
      // registration failed :(
      console.log('ServiceWorker registration failed: ', err);
    });
  });
}
```

- Install:

```
var CACHE_NAME = 'my-site-cache-v1';
var urlsToCache = [
  '/',
  '/styles/main.css',
  '/script/main.js'
];

self.addEventListener('install', function(event) {
  // Perform install steps
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then(function(cache) {
        console.log('Opened cache');
        return cache.addAll(urlsToCache);
      })
  );
});
```

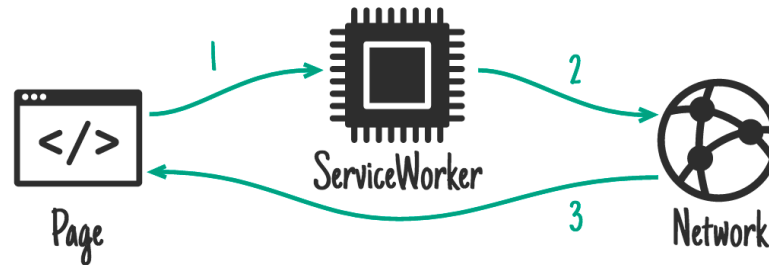

SERVICE WORKER LIFECYCLE

SERVICE WORKER

- **Aufgabe 1:** Es soll ein Service Worker beim Laden der App über die app.js-Datei **registriert** werden. Im Anschluss soll über die DevTools geprüft werden, ob der Service Worker registriert wurde.
- **Aufgabe 2:** Nach der Registrierung des Service Workers soll dieser **installiert** werden. Zusätzlich sollen mit der Installation des Service Worker alle statischen Ressourcen dem **Cache hinzugefügt** (static-cache) werden.

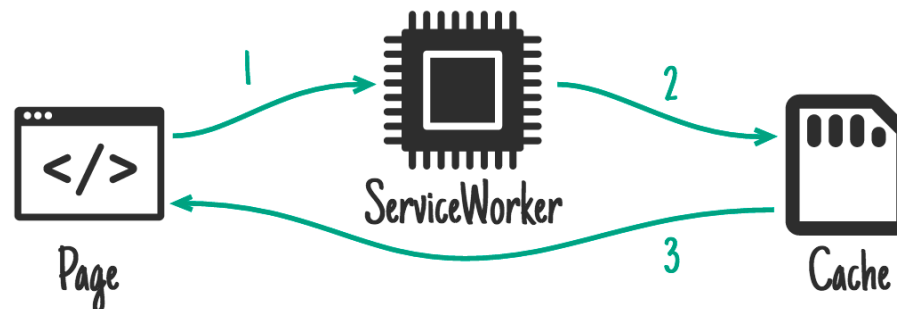
CACHING-STRATEGIEN

- Network only-Strategie



```
self.addEventListener('fetch', function(event) {  
  event.respondWith(fetch(event.request));  
});
```

- Cache only-Strategie

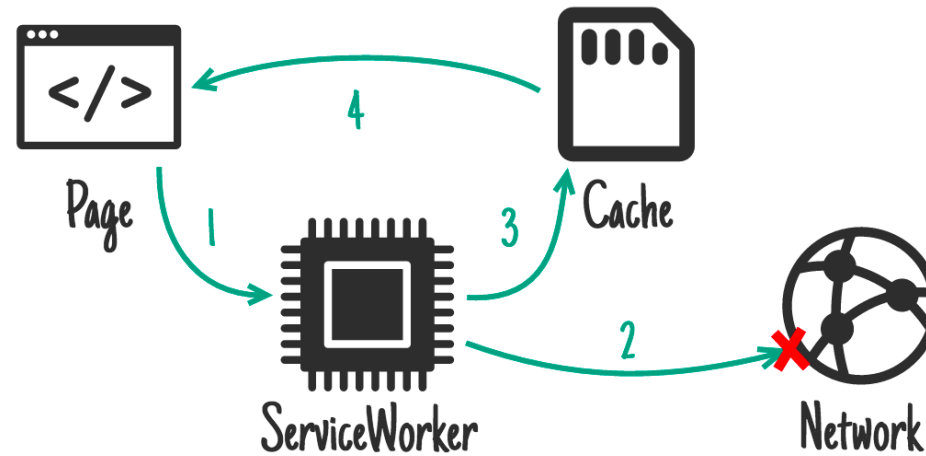


```
self.addEventListener('fetch', function(event) {  
  event.respondWith(caches.match(event.request));  
});
```

- **Aufgabe 3:** Es sollen die statischen Ressourcen der News PWA in einem Offline-Zustand geladen werden. Es soll demnach nur die Benutzeroberfläche angezeigt werden.

CACHING-STRATEGIEN

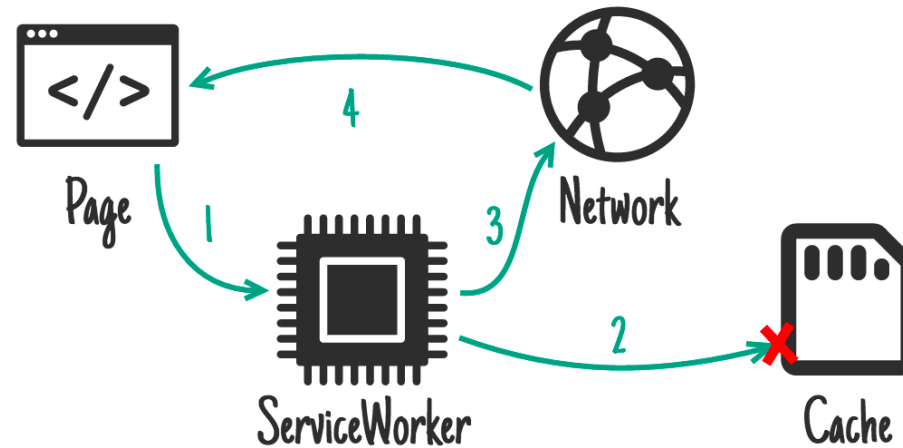
- Network, falling back to cache-Strategie



```
self.addEventListener('fetch', function(event) {
  event.respondWith(
    fetch(event.request).catch(function() {
      return caches.match(event.request);
    })
  );
});
```

CACHING-STRATEGIEN

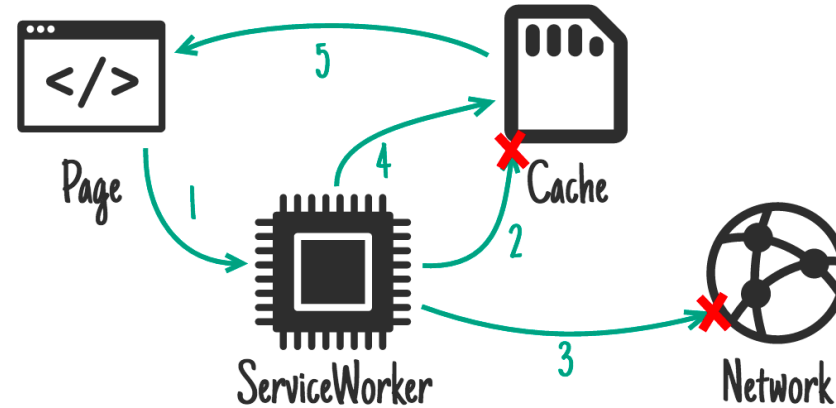
- Cache, falling back to network-Strategie



```
self.addEventListener('fetch', function(event) {
  event.respondWith(
    caches.match(event.request).then(function(response) {
      return response || fetch(event.request);
    })
  );
});
```

CACHING-STRATEGIEN

- Generic Fallback-Strategie



```
self.addEventListener('fetch', function(event) {
  event.respondWith(
    // Try the cache
    caches.match(event.request).then(function(response) {
      // Fall back to network
      return response || fetch(event.request);
    }).catch(function() {
      // If both fail, show a generic fallback:
      return caches.match('/offline.html');
      // However, in reality you'd have many different
      // fallbacks, depending on URL & headers.
      // Eg, a fallback silhouette image for avatars.
    })
  );
});
```

- **Aufgabe 4:** Es soll die **Generic Fallback-Strategie** angewendet werden. Es soll zuerst versucht werden die Ressourcen **aus dem Cache** zu holen.
Gelingt das nicht sollen die Ressourcen **über das Netzwerk angefragt werden** und dem Cache hinzugefügt werden (Dynamischen Ressourcen). Dabei soll ein **neuer Cache** erstellt werden (dynamic-cache) und geöffnet werden.
Scheitert der versuch die Ressource über das Netzwerk anzufragen, soll auf eine **Fallback-Datei** zurückgegriffen werden..

SERVICE WORKER

- **Background Sync** ist eine Web API, mit der Aktionen verschoben werden können, bis der Nutzer über eine stabile Netzwerkverbindung verfügt.
- Nützlich um einen Datenversand bei einer instabilen Netzwerkverbindung an den Server sicherzustellen.
- Auch wenn die Web App nicht im Browser geöffnet ist (Grund: Service Worker, der im Hintergrund läuft und die Synchronisation in einem Service Worker ausgelöst wird)
- **Ablauf:**
 - Nutzer macht eine Anfrage
 - Anstatt die Anfrage sofort zu bearbeiten greift der Service Worker ein
 - Service Worker prüft, ob der Nutzer über eine Internetverbindung verfügt
 - Wenn ja, wird Anfrage gesendet
 - Wenn nicht, dann wartet Service Worker und sendet bei verfügbarer Internetverbindung, nachdem er Daten aus der IndexedDB geholt hat.

BACKGROUND SYNC