



CODERS.BAY

PWAs

CODERS.BAY



PWA



- Progressive-Web-App
- Neue Methode zur App-Entwicklung
- Vor allem durch Google gefördert
- Webseite und App in einem
- Muss nicht installiert werden
- Kann teilweise auf Funktionen des Gerätes zurückgreifen

Vorteile	Nachteile
Plattformunabhängig	PWA-Format noch in Entwicklung
Nicht so aufwendig und kostenintensiv wie native Apps	Nicht alle Browser und Betriebssysteme unterstützen PWAs in vollem Umfang
Zugriff auf: Push-Nachrichten, Geolokalisierung, Kamera, Mikrofon	Kein Zugriff auf Kontakte, Kalender, Bluetooth oder NFC
Offline und online Anwendung	Keine Positionierung im App-Store möglich
Updates automatisch, sobald man online ist	REST-ful APIs bauen
	Indexed DB etwas komplizierter

<https://www.ionos.at/digitalguide/websites/web-entwicklung/progressive-web-apps-welche-vorteile-bieten-sie/>

- SSL – HTTPS
 - Ohne Verschlüsselung funktioniert die PWA nicht
- ServiceWorker (js)
 - Installiert und aktiviert die App
- Manifest (json)
 - Name der App
 - Farben
 - Bilder (Pflicht)

```
let cache = 'my_pwa';
let filesToCache = [
  'index.html',
  'css/main.css',
  'js/main.js',
];

self.addEventListener('install', function(e) {
  //console.log('Install!');
  e.waitUntil(
    caches.open(cache).then(function(cache) {
      //console.log('[ServiceWorker] Caching app shell');
      return cache.addAll(filesToCache).then(() => console.log('Assets added to cache'))
        .catch(err => console.log('Error while fetching assets', err));
    })
  );
});

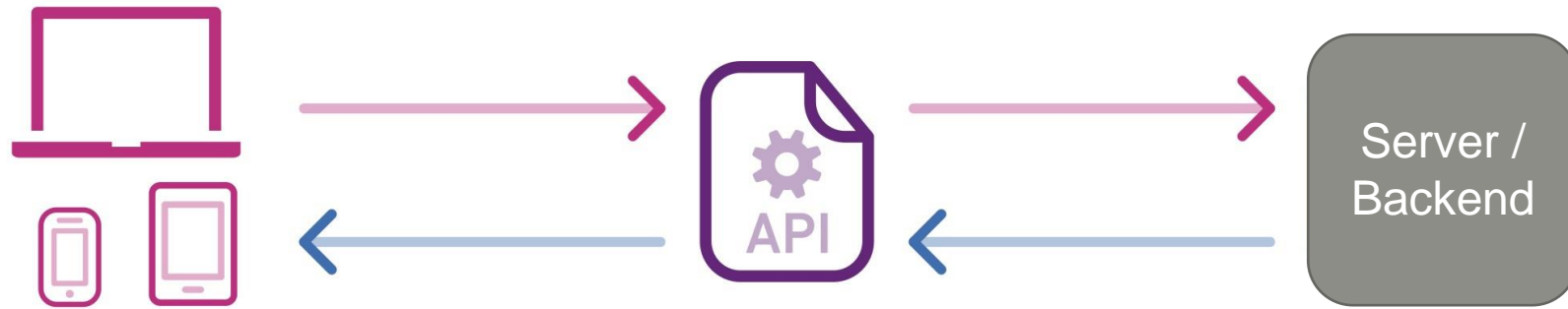
self.addEventListener('activate', event => {
  // console.log('Activate!');
});

self.addEventListener('fetch', function(event) {
  // console.log('Fetch!');
});
```

- <https://app-manifest.firebaseapp.com/>
- start_url – diese Datei wird beim Start aufgerufen
- display
 - browser
 - standalone

```
{
  "short_name": "MyPWA",
  "name": "MyPWA",
  "icons": [
    {
      "src": "/img/logo152.png",
      "sizes": "152x152",
      "type": "image/png"
    },
    {
      "src": "/img/logo192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "/img/logo512.png",
      "sizes": "512x512",
      "type": "image/png"
    },
    {
      "src": "/img/logo900.png",
      "sizes": "900x900",
      "type": "image/png"
    }
  ],
  "start_url": "index.html",
  "background_color": "#A00020",
  "theme_color": "#A00020",
  "display": "standalone"
}
```

API



- **MySQL-Datenbank**
 - Verbindung über PHP & Ajax
- **JSON**
 - Datei liegt am Server
 - Datei wird während der Anwendung gespeichert
- **Offline nicht möglich**
 - Abfrage ob online: navigator.online
- Speicherung in einer **IndexedDB**
 - Sehr aufwändig
 - https://developer.mozilla.org/de/docs/Web/API/IndexedDB_API/Using_IndexedDB



TEMPLATE ENGINE



- Kein PHP - da offline
- Um trotzdem Templates laden zu können
 - Kann man selber schreiben
 - JS Template Engines:

<https://colorlib.com/wp/top-templating-engines-for-javascript/>



REST-API



- Representational State Transfer
 - Architekturstil / Vorgaben um modernes Web sicher zu machen
 - Einheitliche Schnittstelle
- HTTP-Methoden
 - Get
 - Post
 - Put
 - Patch
 - Delete
 - Head
 - Options
 - Connect
 - Trace
- <https://entwickler.de/online/web/restful-api-design-intro-579826380.html>



CODERS.BAY

ENDE