

Offline web applications don't exist anymore

Francesco Leardini

Consultant

francesco.leardini@trivadis.com



@paco_ITA

BASLE ▪ BERN ▪ BRUGG ▪ DÜSSELDORF ▪ FRANKFURT A.M. ▪ FREIBURG I.BR. ▪ GENEVA
HAMBURG ▪ COPENHAGEN ▪ LAUSANNE ▪ MUNICH ▪ STUTTGART ▪ VIENNA ▪ ZURICH

trivadis
makes IT easier. ■ ■ ■

- Once upon a time there was the *downasaur*...



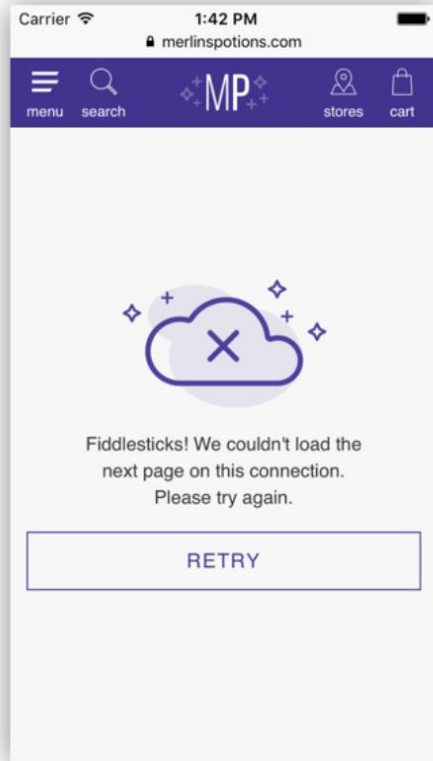
No internet

Try:

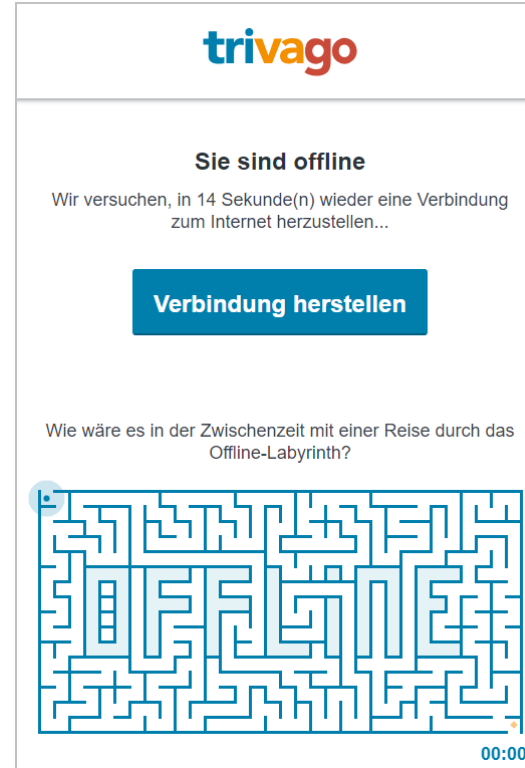
- Checking the network cables, modem, and router
- Reconnecting to Wi-Fi
- [Running Windows Network Diagnostics](#)

DNS_PROBE_FINISHED_NO_INTERNET

Modern websites nowadays!



Branded offline pages



Engaging pages

■ PWA Stats

The **Best Western River North Hotel** sees **300%** increase in revenue with new Progressive Web App.



PWA Stats

Twitter Lite saw a **65%** increase in pages per session, **75%** in Tweets, and a **20%** decrease in bounce rate. Twitter Lite loads in under **3** seconds for repeat visits even on slow networks.



AliExpress improved conversion rate for new users by **104%** across all browsers, with **2x** more pages visited and **74%** more time spent per session.



www.pwastats.com

■ What is a Progressive Web Application (PWA)

A PWA is a web app using modern web capabilities to provide a native app experience to users

- **Installable:** thanks to the web app manifest
- **Work offline:** using service workers (sw)
- Can send **push notifications**

■ Web App Manifest (manifest.json)



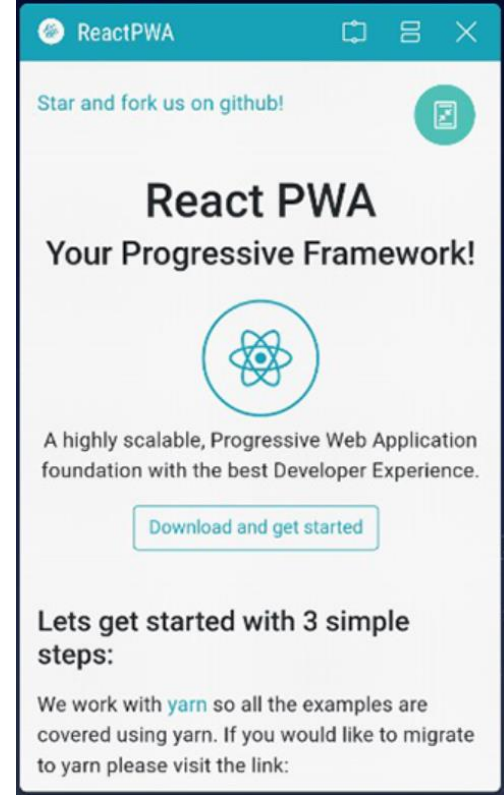
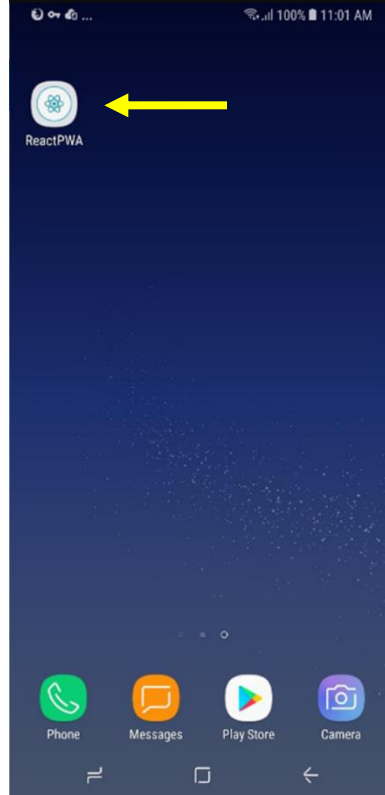
Chrome criteria

Https

Registered SW

```
1  {
2  "name": "My PWA example",
3  "short_name": "pwa-app",
4  "description": "A simple description field!",
5  "display": "standalone",
6  "start_url": "/",
7  "icons": [
8    {
9      "src": "assets/icons/icon-192x192.png",
10     "sizes": "192x192",
11     "type": "image/png"
12   },
13   {
14     "src": "assets/icons/icon-512x512.png",
15     "sizes": "512x512",
16     "type": "image/png"
17   }
18 ]
19 }
```

■ Add to Home Screen (A2HS)



■ How does a PWA makes websites available offline?

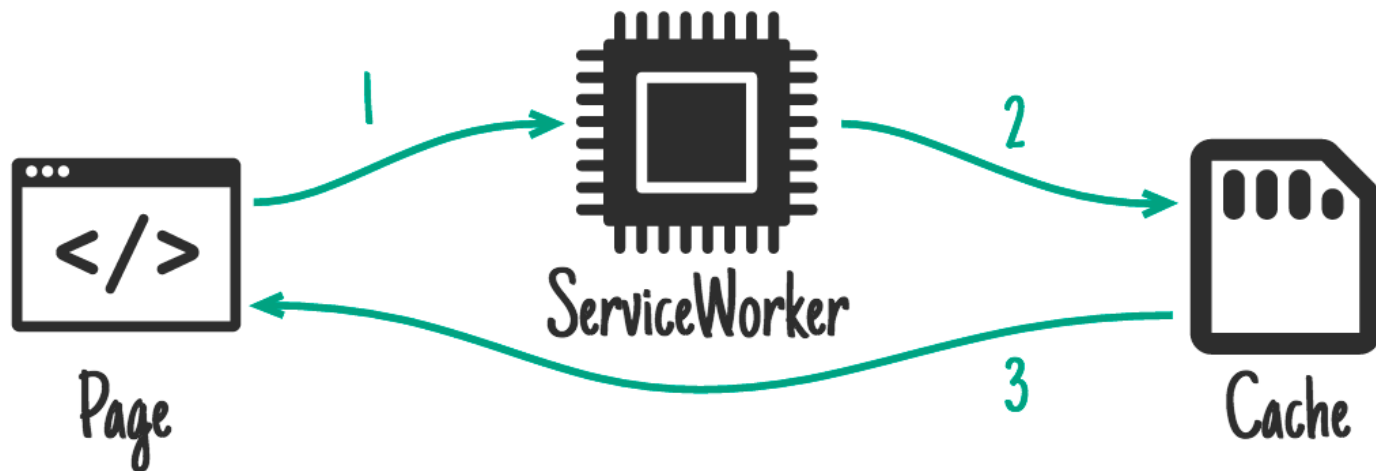
A **service worker**:

- is a component of JavaScript code
- works as a proxy between the browser and the network
- runs in the background on a separate thread (no blocking)
- works only on secure connections (localhost is considered secure)

■ Service worker caching strategies for data resources

Performance optimization (cache only)

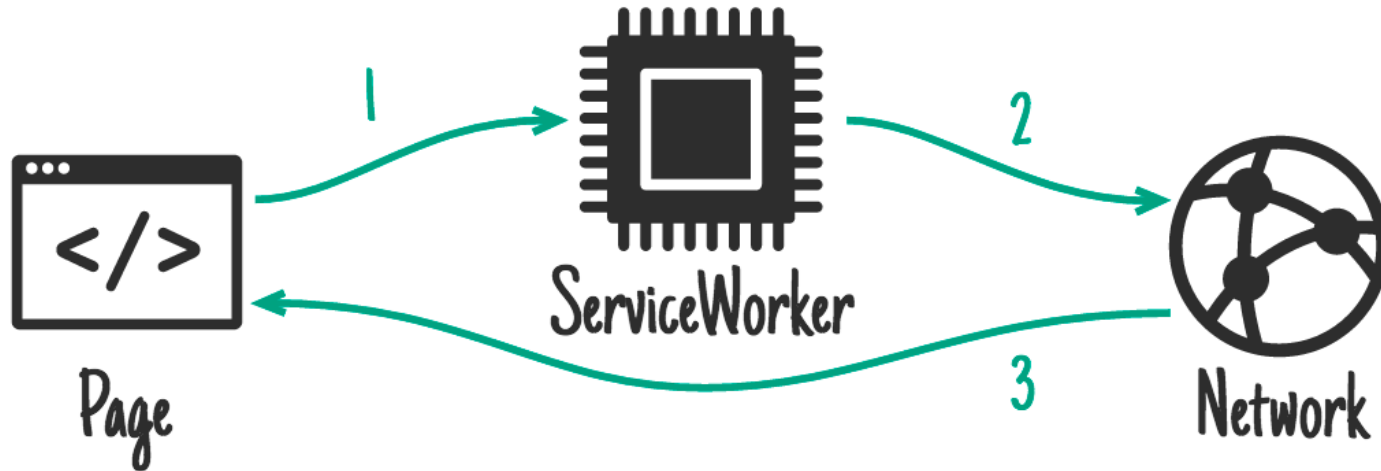
→ Needed data has to be previously cached (typically during the sw install phase)



■ Service worker caching strategies for data resources

Data freshness

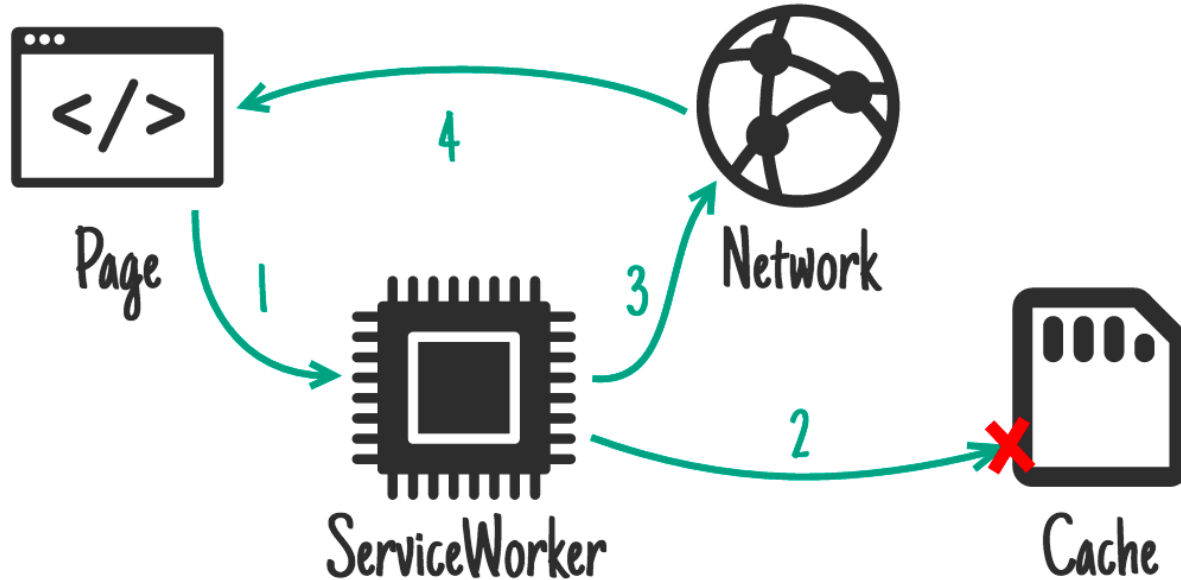
→ Useful for things that have no offline equivalent, such as analytics pings



■ Service worker caching strategies for data resources

Cache, falling back to network

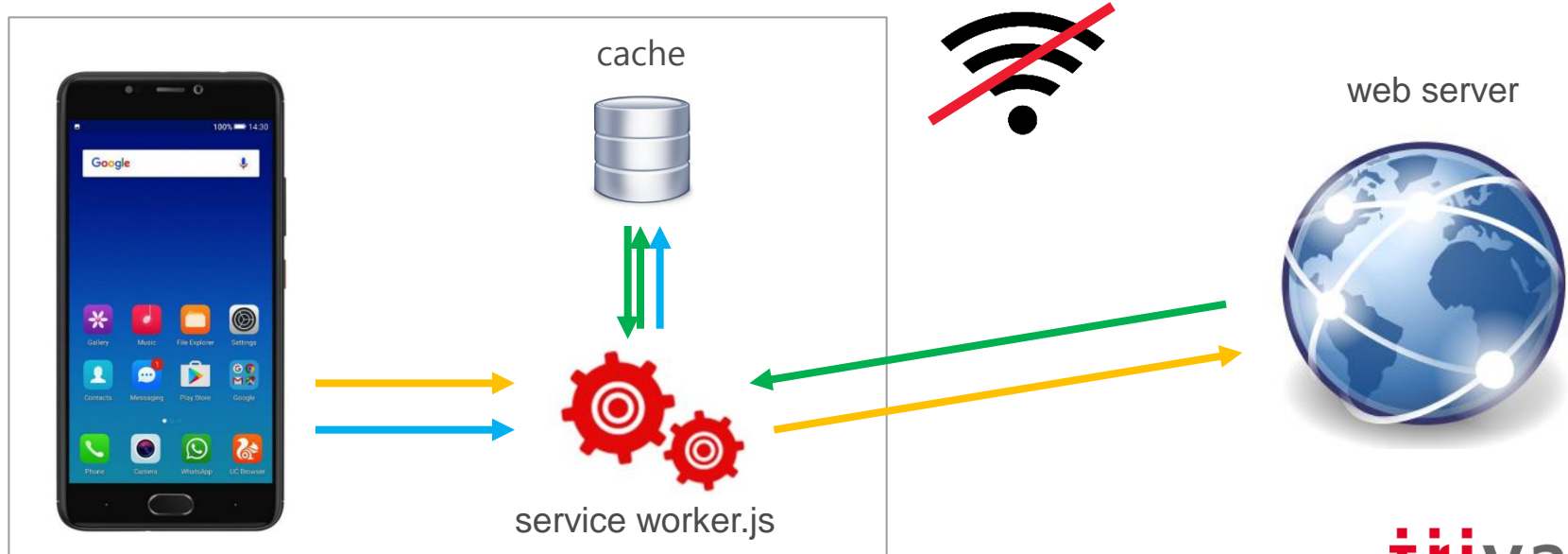
→ Ideal for the majority of requests in offline-first apps



■ Service worker in action

The first time the app is accessed, the service worker stores the required resources in the browser cache.

When the user visits the app next time, the service worker checks the cache and returns the response to the user even before accessing the web server.





PWA BUILDER

1

Generate Manifest

2

Build Service Worker

3

Publish PWA

Select the service worker functionality for your app.

Offline page

This simple but elegant solution pulls a file from your web server called "offline.html" (make sure that file is actually there) and serves the file whenever a network connection can not be made.

Offline copy of pages

A solution that expands the offline capabilities of your app. A copy of each pages is stored in the cache as your visitors view them. This allows a visitor to load any previously viewed page while they are offline.

Offline copy with Backup offline page

A copy of each pages is stored in the cache as your visitors view them. This allows a visitor to load any previously viewed page while they are offline. This then adds the "offline page" that allows you to customize the message and experience if the app is offline, and the page is not in the cache.

Cache-first network

Use this service worker to pre-cache content. The content you add to the "cache-array" will be added immediately to the cache and service from the cache whenever a page requests it. At the same time it will update the cache with the version you have on the server. Configure your file array to include all your site files, or a subset that you want to be served quickly.

Advanced Pre-cache (coming soon)

Use this service worker to improve the performance of your app, and make it work offline. The advanced pre-cache allows you to configure files and routes that are cached in different manners (pre-cache, server first, cache first etc). The tool can be used to build a lightning fast app (even for dynamic content) that works offline.

Download service worker

You can also download the source code directly from [GitHub](#).

Code for website

Next Step

```
//This is the "Offline page" service worker

//Add this below content to your HTML page, or add the js file to your page at the very top to
register service worker
if (navigator.serviceWorker.controller) {
  console.log('[PWA Builder] active service worker found, no need to register')
} else {
  //Register the ServiceWorker
  navigator.serviceWorker.register('pwabuilder-sw.js', {
    scope: './'
  }).then(function(reg) {
    console.log('Service worker has been registered for scope:'+ reg.scope);
  });
}
```

Service worker code

```
//This is the "Offline page" service worker

//Install stage sets up the offline page in the cache and opens a new cache
self.addEventListener('install', function(event) {
  var offlinePage = new Request('offline.html');
  event.waitUntil(
    fetch(offlinePage).then(function(response) {
      return caches.open('pwabuilder-offline').then(function(cache) {
        console.log('[PWA Builder] Cached offline page during Install'+ response.url);
        return cache.put(offlinePage, response);
      });
    });
});
```



Cache Images

```
1 workbox.routing.registerRoute(  
2   /\.(?:png|gif|jpg|jpeg|svg)$/,  
3   workbox.strategies.cacheFirst({  
4     cacheName: 'images',  
5     plugins: [  
6       new workbox.expiration.Plugin({  
7         maxEntries: 60,  
8         maxAgeSeconds: 30 * 24 * 60 * 60, // 30 Days  
9       }),  
10    ],  
11  }),  
12 );
```



ment

uild a progressive web app? Workbox makes
ate an offline first experience.

■ Service Workers in Angular



With angular-cli (1.6+) we can easily turn an existing application into a PWA:

```
ng add @angular/pwa --project myApp
```

Two new files will be created and added to our project:

- ngsw-config.json
- manifest.json

Angular PWA demo

Can we do better?

■ Firebase platform



Build better apps



Cloud Firestore BETA

Store and sync app data at global scale



ML Kit BETA

Machine learning for mobile developers



Cloud Functions BETA

Run mobile backend code without managing servers



Authentication

Authenticate users simply and securely



Hosting

Deliver web app assets with speed and security



Cloud Storage

Store and serve files at Google scale



Realtime Database

Store and sync app data in milliseconds



Improve app quality



Crashlytics

Prioritize and fix issues with powerful, realtime crash reporting



Performance Monitoring

Gain insight into your app's performance



Test Lab

Test your app on devices hosted by Google



Grow your business



Google Analytics

Get free and unlimited app analytics



Predictions BETA

Define dynamic user groups based on predicted behavior



A/B Testing BETA

Optimize your app experience through experimentation



Cloud Messaging

Send targeted messages and notifications



Remote Config

Modify your app without deploying a new version



Dynamic Links

Drive growth by using deep links with attribution



App Indexing

Drive search traffic to your mobile app

■ Cloud Firestore

- Still in beta release (Oct 2017)
- NoSQL Database
- Real-time data updates (*3-way binding*)
- **Offline support** for mobile (iOS/Android) and web

■ Enable offline data in Cloud Firestore

```
firebase.initializeApp({
  apiKey: '### FIREBASE API KEY ###',
  authDomain: '### FIREBASE AUTH DOMAIN ###',
  projectId: '### CLOUD FIRESTORE PROJECT ID ###',
})

firebase.firestore().enablePersistence()
.catch(function(err) {
  if (err.code == 'failed-precondition') {
    // Multiple tabs open, persistence can only be enabled
    // in one tab at a a time.
    // ...
  } else if (err.code == 'unimplemented') {
    // The current browser does not support all of the
    // features required to enable persistence
    // ...
  }
});
// Subsequent queries will use persistence, if it was enabled successfully
```

■ Offline data limitations

- ~~Only one browser tab per time~~ → **version 5.5.0** (12.09.2018)
- Max 500 offline transactions possible
- Possible saving conflicts (*last write wins*)

A «*never offline*» app demo

Questions?



BASLE ▪ BERN ▪ BRUGG ▪ DÜSSELDORF ▪ FRANKFURT A.M. ▪ FREIBURG I.BR. ▪ GENEVA
HAMBURG ▪ COPENHAGEN ▪ LAUSANNE ▪ MUNICH ▪ STUTTGART ▪ VIENNA ▪ ZURICH

trivadis
makes IT easier. ■ ■ ■

Francesco Leardini

Consultant

francesco.leardini@trivadis.com



Trivadis
makes IT
easier.



@paco_ITA

BASLE ▪ BERN ▪ BRUGG ▪ DÜSSELDORF ▪ FRANKFURT A.M. ▪ FREIBURG I.BR. ▪ GENEVA
HAMBURG ▪ COPENHAGEN ▪ LAUSANNE ▪ MUNICH ▪ STUTTGART ▪ VIENNA ▪ ZURICH

trivadis
makes IT easier. ■ ■ ■