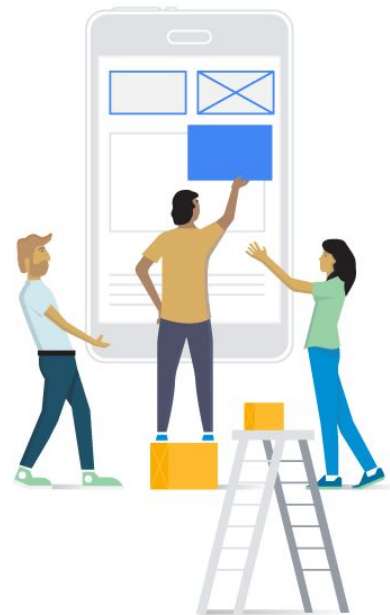




Welcome to the **PWA** webinar



Why PWA?



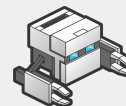
The 2 mobile channels



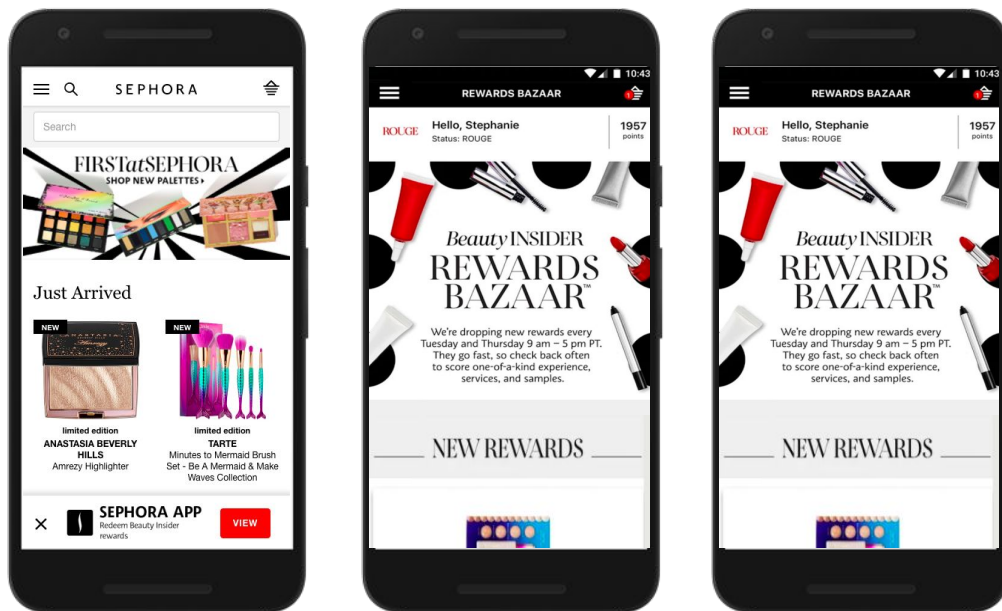
Mobile Web



Mobile apps



One site + 2 apps



One + 2 native apps



3 teams

3 projects

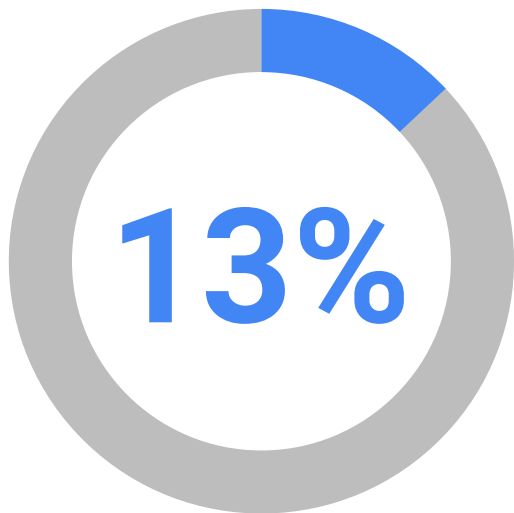
3 technologies

3 times more meetings

3 times more code to maintain

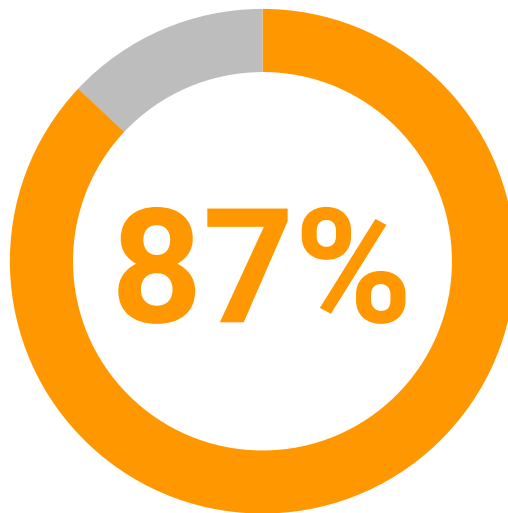
To do the same thing...

Most of the time spent on apps



Mobile web

VS.



Mobile apps

Source: comScore Mobile Metrix, U.S., Age 18+, Juin 2016

Why ?

Mobile apps



Mobile first

Mobile design



Fast

Local assets



Visibles

Home screen



Engaging

Push notifications

Source: comScore Mobile Metrix, U.S., Age 18+, June 2016

mWeb + App mobile = PWA



Mobile Web



Mobile Apps



PWA

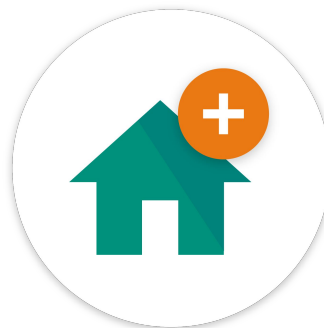
Add these features to your site



Fast



Available
Offline



Add to
home screen



Push
Notifications



Demo

18:13



<https://www.tajawal.com/mweb>



tajawal:

Go beyond



FLIGHTS



HOTELS



HOLIDAYS



Ajouter Tajawal à l'écran d'accueil



18:13



https://www.tajawal.com/mweb



tajawal:

Go beyond

Ajouter à l'écran d'accueil



Tajawal
www.tajawal.com

ANNULER AJOUTER





tajawal:

Go beyond



FLIGHTS



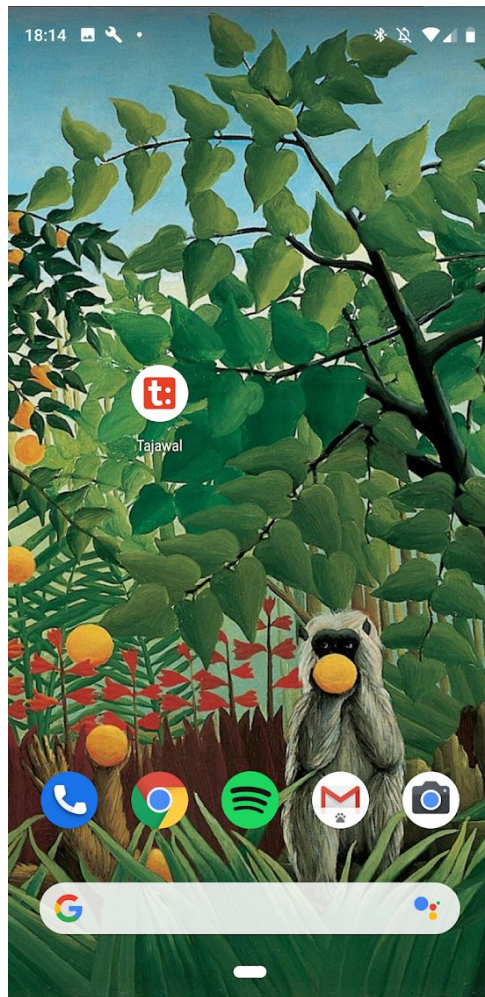
HOTELS



HOLIDAYS

Ajout de Tajawal...





18:15



t:

Tajawal



18:19



5G H+  



tajawal:

Over 1 million hotels at your fingertips!

Enter city, area or hotel name



New York



Check-in

Wed, 17 Apr

Check-out

Thu, 18 Apr



Guests

1 Room, 2 Adults

Recent searches

New York

Wed, 17 Apr - Thu, 18 Apr

Paris

Wed, 17 Apr - Thu, 18 Apr

Search Hotels



PWA

Add to Home screen



Add to Home Screen



Implementation time: 1H

-  1) Create a manifest.json file
-  2) Create icons
-  3) Register a service worker

Add to Home Screen



```
<link rel=manifest href=/static/manifest.json>
```

```
{
  name: "vue-todo",
  short_name: "ToDo",
  - icons: [
    - {
      src: "/static/img/icons/android-chrome-192x192.png",
      sizes: "192x192",
      type: "image/png"
    },
    - {
      src: "/static/img/icons/android-chrome-512x512.png",
      sizes: "512x512",
      type: "image/png"
    }
  ],
  start_url: "/index.html",
  display: "standalone",
  background_color: "#000000",
  theme_color: "#4DBA87"
}
```

Web App Manifest Generator

The [Web App Manifest](#) is a JSON document that provides application metadata for [Progressive Web Apps](#). Use the form below to generate the JSON file and optionally upload an app icon.

App Name

Placeholder

Short Name

Placeholder

Theme Color

#2196f3

Background Color

#2196f3

Display Mode

Browser

Orientation

Any

manifest.json

```
{
  "theme_color": "#2196f3",
  "background_color": "#2196f3",
  "display": "browser",
  "scope": "/",
  "start_url": "/"
}
```

COPY

Generate Icons

The Web App Manifest allows for specifying icons of varying sizes. Upload a 512x512 image for the icon and we'll generate the remaining sizes.

ICON

GENERATE .ZIP

<https://app-manifest.firebaseapp.com/>

Icons on IOS

```
<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-mobile-web-app-status-bar-style" content="black">
<meta name="apple-mobile-web-app-title" content="My App Name">

<link rel="apple-touch-icon" href="/src/images/icons/apple-icon-57x57.png" sizes="57x57">
<link rel="apple-touch-icon" href="/src/images/icons/apple-icon-60x60.png" sizes="60x60">
<link rel="apple-touch-icon" href="/src/images/icons/apple-icon-72x72.png" sizes="72x72">
<link rel="apple-touch-icon" href="/src/images/icons/apple-icon-76x76.png" sizes="76x76">
<link rel="apple-touch-icon" href="/src/images/icons/apple-icon-114x114.png" sizes="114x114">
<link rel="apple-touch-icon" href="/src/images/icons/apple-icon-120x120.png" sizes="120x120">
<link rel="apple-touch-icon" href="/src/images/icons/apple-icon-144x144.png" sizes="144x144">
<link rel="apple-touch-icon" href="/src/images/icons/apple-icon-152x152.png" sizes="152x152">
<link rel="apple-touch-icon" href="/src/images/icons/apple-icon-180x180.png" sizes="180x180">
```

Add icons for IOS ↗

**To earn the right
to be on the home screen**
Is mandatory to have
A **minimal offline support**

PWA

Basic offline support



Offline support

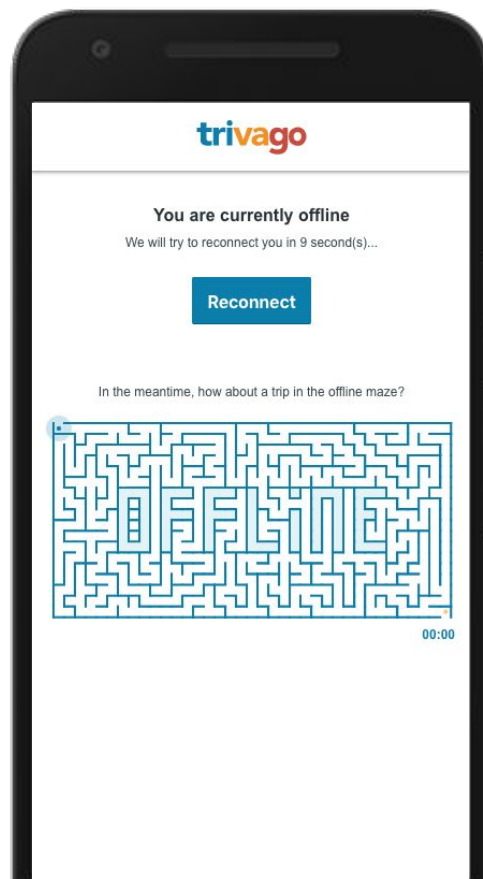
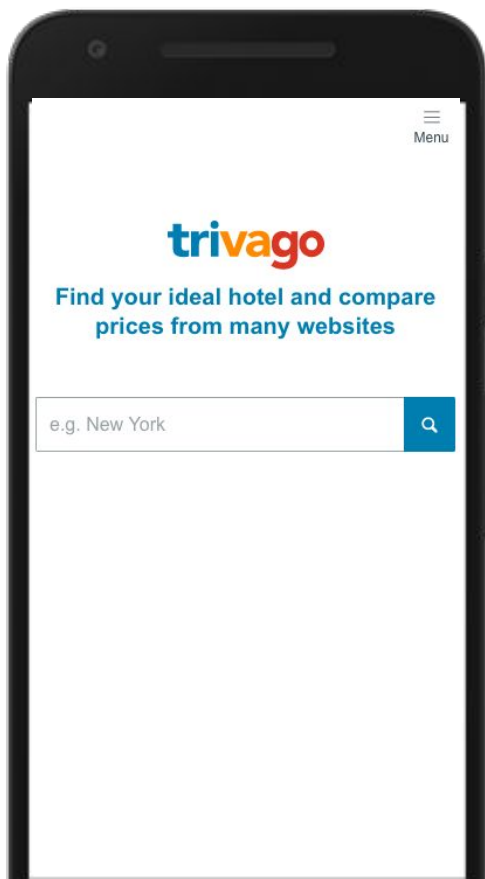


Having the whole site offline is not always necessary.

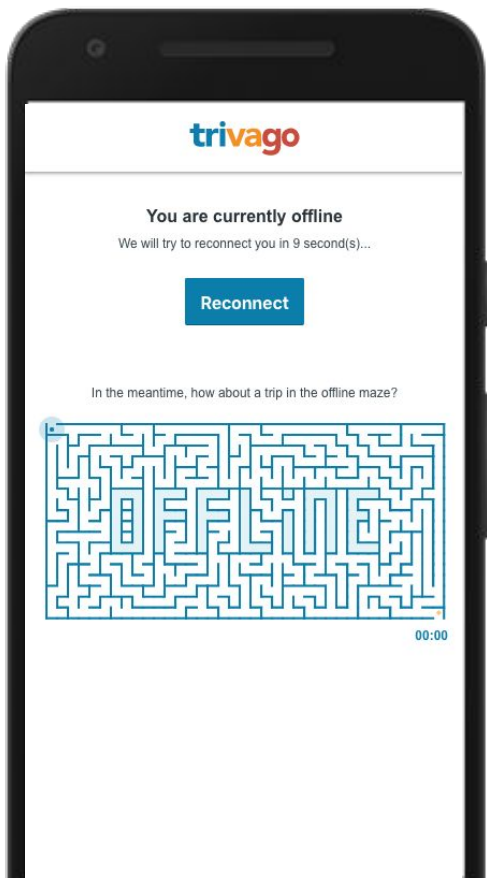


Do native apps work 100% offline?

Offline support



Offline support



Implementation time: 2 H



1) Create a service worker

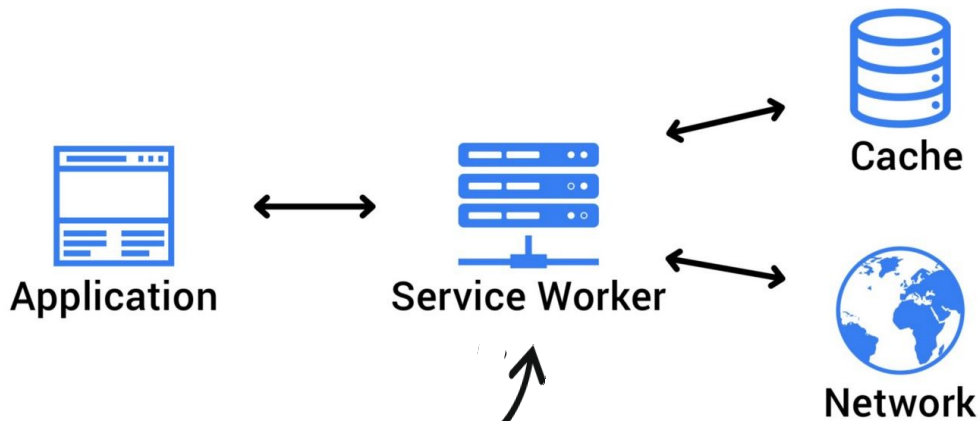


2) Create an offline page



3) Register the service worker

Service worker? What's that?



*Can intercept and
route requests...
Can listen events...*



EXPLORER

▶ OPEN EDITORS

◀ PWA-WORKSHOP

▶ css

▶ img

▶ js

▶ node_modules

◆ .gitignore

<> index.html

{ } manifest.json

{ } package-lock.json

{ } package.json

JS sw.js

JS workbox-config.js



Service worker has special rules

The max scope of a service worker is the service worker's location.

A service worker can only handle requests for clients under its scope.

I put the Service Worker at the root level

A service worker at the root level



```
cd ~/my-project/
```

```
touch sw.js
```



I created a Service Worker at the root of my project

A service worker at the root level



```
# Service Worker ! No cache
<filesMatch serviceworker.js>
FileETag None
Header unset ETag
Header set Cache-Control "max-age=0, no-cache, no-store, must-revalidate"
Header set Pragma "no-cache"
Header set Expires "Wed, 11 Jan 1984 05:00:00 GMT"
</filesMatch>
```



*Be sure to not cache your
Service Worker*

A service worker at the root level



```
<IfModule mod_rewrite.c>  
RewriteEngine On  
RewriteRule serviceworker.js mycoolSite/path/assets/serviceworker.js [L]  
</IfModule>
```

↖ You can always serve you SW
at root level but have it in
your project's folder

code

Listening events in the Service Worker

*In my sw.js
I listen for fetch events*

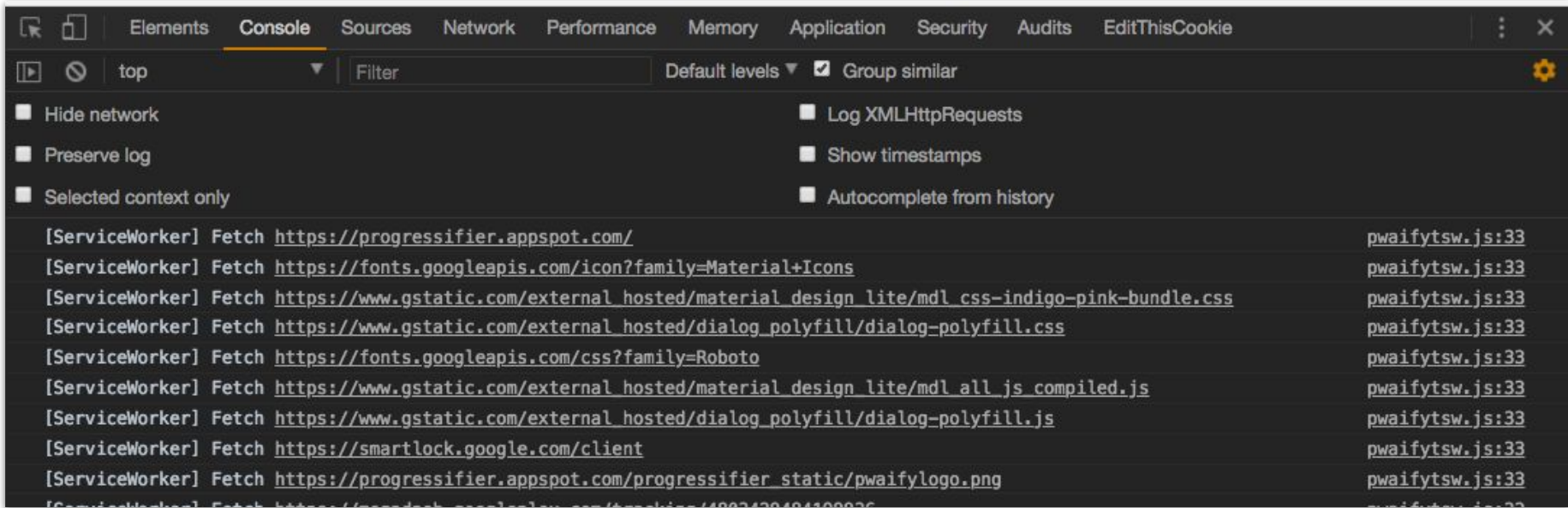
```
self.addEventListener("fetch", event => console.log(`[ServiceWorker] Fetch ${event.request.url}`))
```


Registering the Service Worker

↖ In my app.js

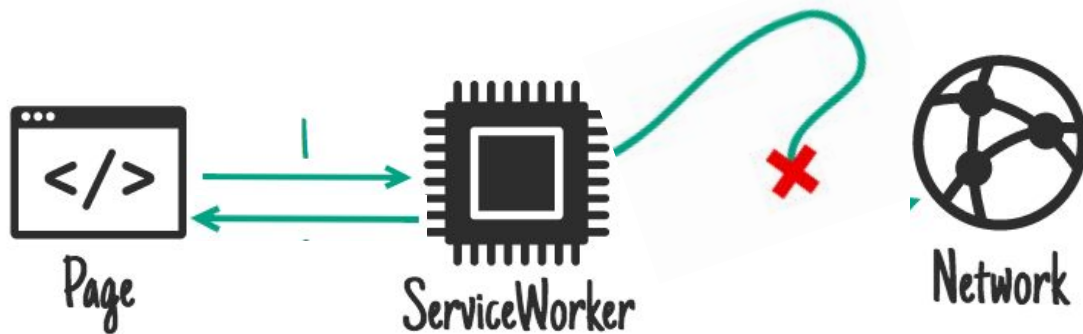
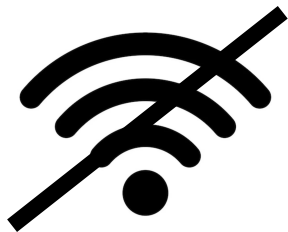
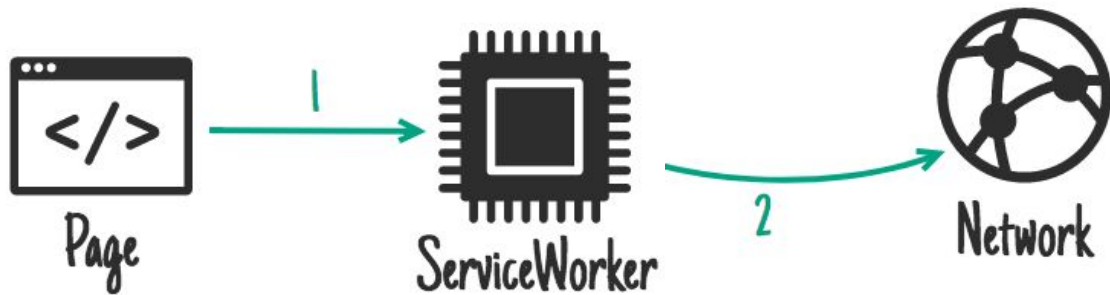
```
(async () => {  
  if ("serviceWorker" in navigator) {  
    // we register our service worker  
    const registration = await navigator.serviceWorker.register('/sw.js');  
  
    // when our service worker is updated  
    registration.onupdatefound = () => {  
      // when our service worker is updated  
      registration.installing.onstatechange = function () {  
        console.log(`Service worker... ${this.state}`);  
      };  
    };  
  }  
})();  
.catch(e => console.log(`🤖 : ${e}`));
```

*The Service Worker
logs the requests*



He can **intercept**
and modify
requests in our Service
Worker

A basic offline page



The most basic offline page

```
self.addEventListener("fetch", event => {  
  event.respondWith(  
    fetch(event.request)  
    .catch( () => new Response("<h1>You are offline</h1>", { headers : {"Content-Type": "text/html"}}))  
  );  
});
```



*We listen the fetch event
If it fails we send a custom response*

Hi there!

I'm a basic PWA

I'm installable and I will show an offline page :)



Elements Network Console Sources Performance Memory Application Security Audits >>

View: [Icons] Group by frame [] Preserve log [x] Disable cache [] Offline Online

Filter [] Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other

Name	Method	Status	Domain	Type	Initi...	Size	Time	C...	Waterfall
mybasicpwa.glitch.me	GET	200	mybasicpwa....	document	Other	(from Servic...	169 ms		

1 / 12 requests | 0 B / 71.7 KB transferred | Finish: 1.80 s | DOMContentLoaded: 434 ms | Load: 654 ms

Not bad but..

We can do better

```
const offlineHTML = `
```

```
<!DOCTYPE html>
<html class="no-js">

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>My site | Offline </title>
  <meta name="description" content="">
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>

<style>
  body {
    font-family: "Benton Sans", "Helvetica Neue", helvetica, arial, sans-serif;
    margin: 2em;
  }

  h1 {
    font-style: italic;
    color: #373fff;
  }

  img{
    display: block;
    margin: auto;
  }
</style>

<body>
  <h1>You are offline 📴 </h1>
</body>

</html>
```

```
`;
```

```
self.addEventListener("fetch", event => {

  event.respondWith(
    fetch(event.request)
      .catch( () => new Response(offlineHTML, { headers : {"Content-Type": "text/html;charset=utf-8"}}))
  );

});|
```

Let's create an
offline HTML inside
our sw.js

Service worker? What's that?

*When offline we send back
offlineHTML*

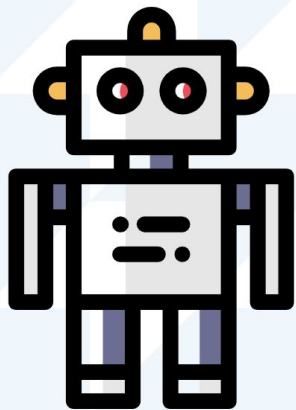


```
const offlineHTML = `
`;

self.addEventListener("fetch", event => {
  event.respondWith(
    fetch(event.request)
      .catch( () => new Response(offlineHTML, { headers : {"Content-Type": "text/html;charset=utf-8"}}))
  );
});
```

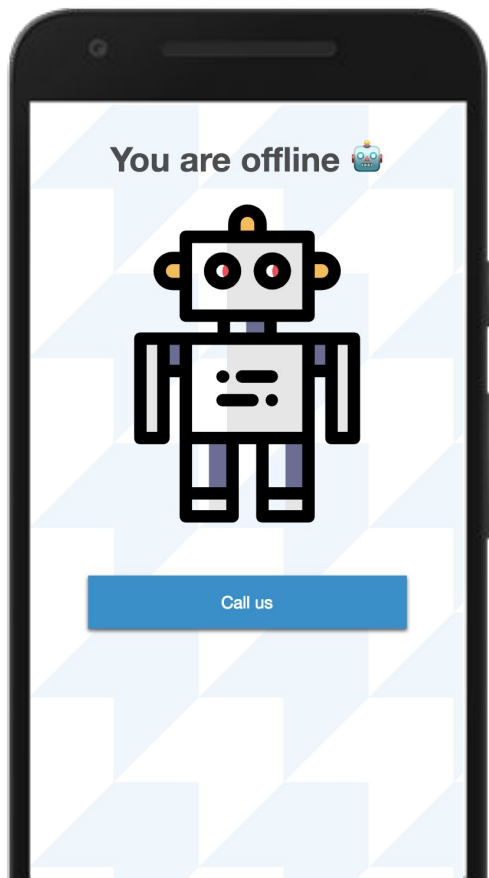
Name	Method	Status	Domain	Type	Initi...	Size	Time	C...	Waterfall	▲
mybasicpwa.glitch.me	GET	200	mybasicpwa....	document	Other	(from Servic...	33 ms			

You are offline 🤖



Call us

It's your time to try : <https://pwa101.glitch.me/>



Implementation time: 30min



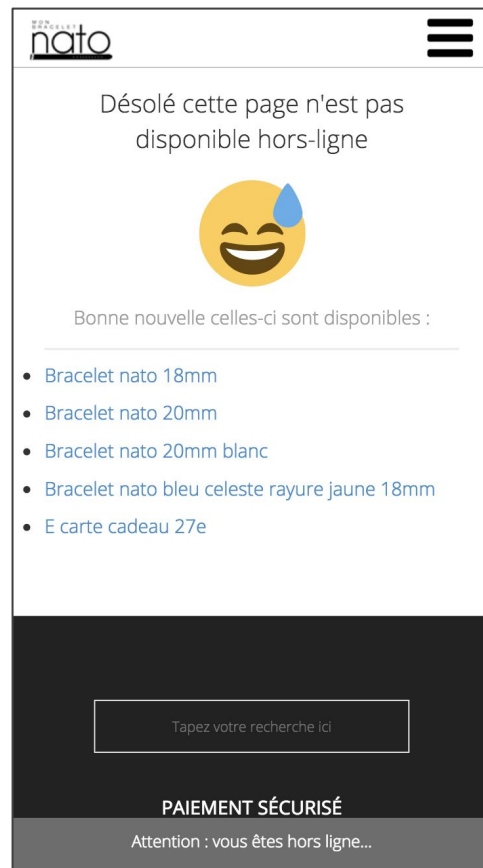
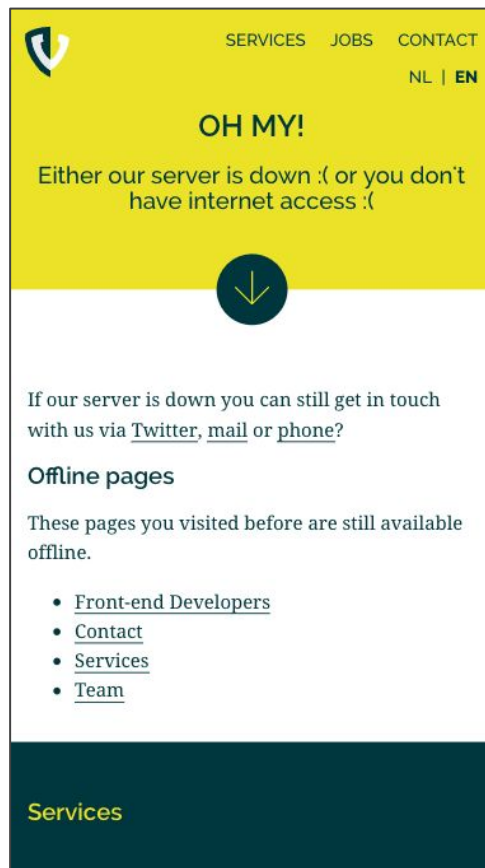
1) Create a service worker



2) Create an offline page



3) Register the service worker



PWA

Advanced concepts



Beyond a basic offline page with the **Cache API**

I'm A Basic PWA

I'm installable and I will show an offline page :)



Subscribe to Notifications

Application

- Manifest
- Service Workers
- Clear storage

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies

Cache

- Cache Storage
- Application Cache

Frames

- top

Different ways to store data client's side



1) Local Storage



2) IndexedDb



3) Web SQL



4) Cache API



5) App cache

The cache API

The Cache API is a system for **storing and retrieving network requests** and corresponding responses. The API can be accessed from a window, iframe, worker, or service worker.

The cache API : create a cache

← *cache.open()
will open or create a cache*

```
cache.open("my-pwa-demo").then( cache => {  
  console.log(`The created cache : ${cache}`);  
})
```

I'm An Advanced PWA

I'm installable and I will show an offline page :)



Application

- Manifest
- Service Workers
- Clear storage

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies

Cache

- Cache Storage
 - my-pwa-demo - <https://zealous-wave.glitch.u>
 - Application Cache

Frames

- top

← The cache we created

The cache API : Let's add assets to our cache

*We wait for our SW
to be installed*

```
self.addEventListener('install', async event => {  
  console.log('[SW] installing SW', event);  
  
  const myPWAdemoCache = await caches.open("my-pwa-demo");  
  
  await myPWAdemoCache.addAll(['/', 'index.html', 'script.js', 'style.css']);  
});
```

code

I'm An Advanced PWA

I'm installable and I will show an offline page :)



Application

- Manifest
- Service Workers
- Clear storage

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies

Cache

- Cache Storage
 - my-pwa-demo - <https://zealous-wave.glitch.>
- Application Cache

Frames

- top

Path	Content-Type
/index.html	text/html; charset=utf-8
/script.js	application/javascript; charset=utf-8
/style.css	text/css; charset=utf-8

Headers

Preview

```
1 /* CSS files add styling rules to your
2
3 body {
4   font-family: "Benton Sans", "Helvetica", "Arial", sans-serif;
5   margin: 2em;
6 }
7
8 h1 {
9   font-style: italic;
10  color: #373fff;
11 }
12
13 img{
14   display:block;
15   margin:auto;
16 }
17
18 .title{
19   display:block;
20   margin:auto;
21   background-color: #ff2121;
22   height: 40px;
23   color: #FFF;
24   line-height: 40px;
25   text-transform: capitalize;
26   text-align: center;
27   font-size: 20px;
28   font-family: monospace;
29   font-weight: 100;
30   transform: skewY(-10deg);
31   margin-bottom: 41px;
32   max-width: 300px;
33 }
34
35
36 p{
37   text-align:center;
38 }
39
40 .btn{
41   height: 50px;
42   display: block;
43   margin: auto;
44 }
```



Our assets
are now in
the cache

Let's respond to requests from the cache API



- 1) Listen for requests in our Service Worker
Open the cache



- 2) if the requests is in our opened cache :
 - Respond from the cache



Respond from cache



- 3) If the request is not in the cache :
 - We fetch the request on the network
 - We respond with the fetched request



Respond from network

Let's respond to requests from the cache API



```
self.addEventListener('fetch', event => {  
  event.respondWith(cacheOrNetwork(event, "my-pwa-demo"))  
});
```



*We listen fetch events
and look inside our
cache to respond*

Let's **respond** to requests
from the **cache API**



```
const cacheOrNetwork = async (event, cacheName) => {  
  
  // Open the cache  
  const cache = await caches.open(cacheName);  
  // check if the requests is in our cache :  
  const responseFromCache = await cache.match(event.request);  
  
  // if we have the request in the cache  
  if(responseFromCache) {  
  
    console.log(`[SW] I'm responding to ${event.request.url} from the cache !!`);  
  
    return responseFromCache;  
  
  }  
}
```

```
// If the request is not in the cache :  
  
// we can't use the request twice we have to clone it !  
const requestToFetch = event.request.clone();  
  
// we fetch the request on the network  
const responseFromNetwork = await fetch(requestToFetch);  
  
// we respond with the fetched request  
return responseFromNetwork;  
  
}
```



*If the request was cached
we respond from cache*



*If the request was NOT cached
we respond from the network*

I'm An Advanced PWA

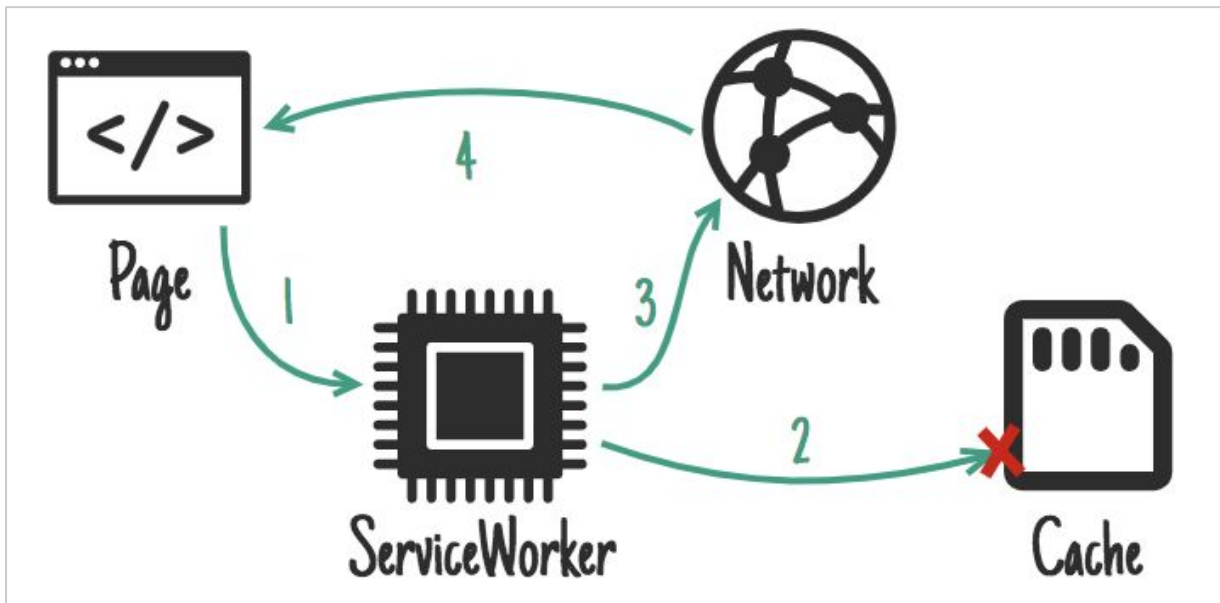
I'm installable and I will show an offline page :)



Recording network activity...

Perform a request or hit ⌘ R to record the reload.

The strategy we just created :



But what if I want to :

Refresh the cache

**Always respond from
network**

Set expires in cache

Update my cache in the
background

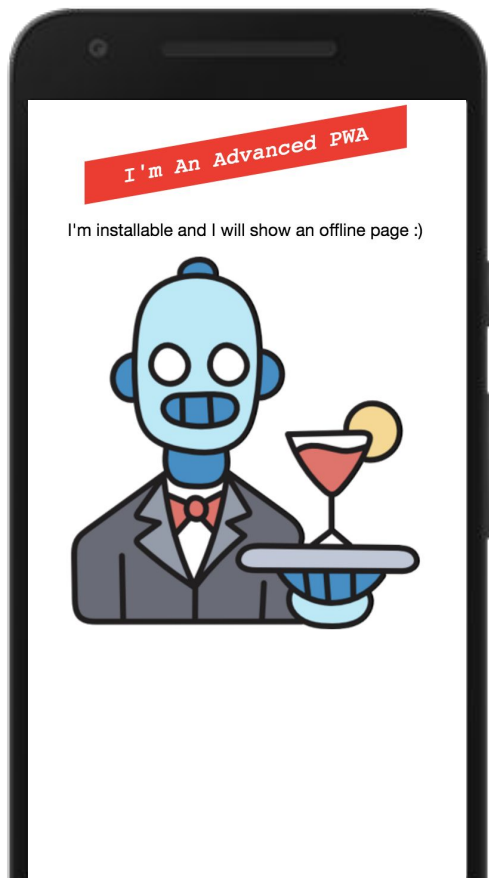
Automatic SW generation
and **easy caching strategies** with



Workbox

*High level API for
Service Worker* ↗

Have a look on the advanced example



Have a look :

<https://myadvancedpwa.glitch.me/>

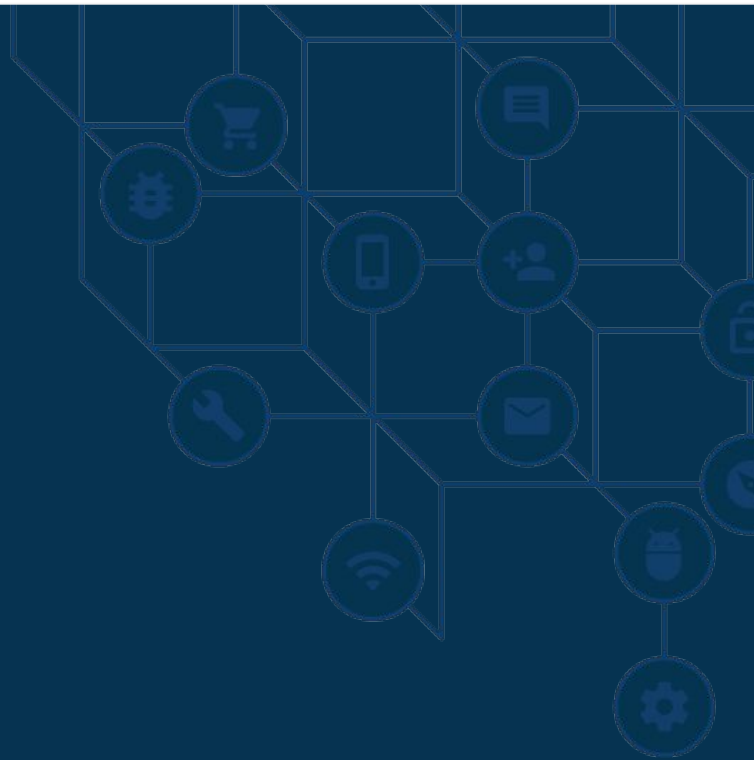
Check the demo 

PWA

Advanced PWA concepts with :



Workbox





Bitcoin

3564 €

2018-12-10T17:00:00Z



Dash

69 €

2018-12-10T17:00:00Z



ethereum

Ethereum

90 €

2018-12-10T17:00:00Z

← We have a basic page which
fetches an API through AJAX

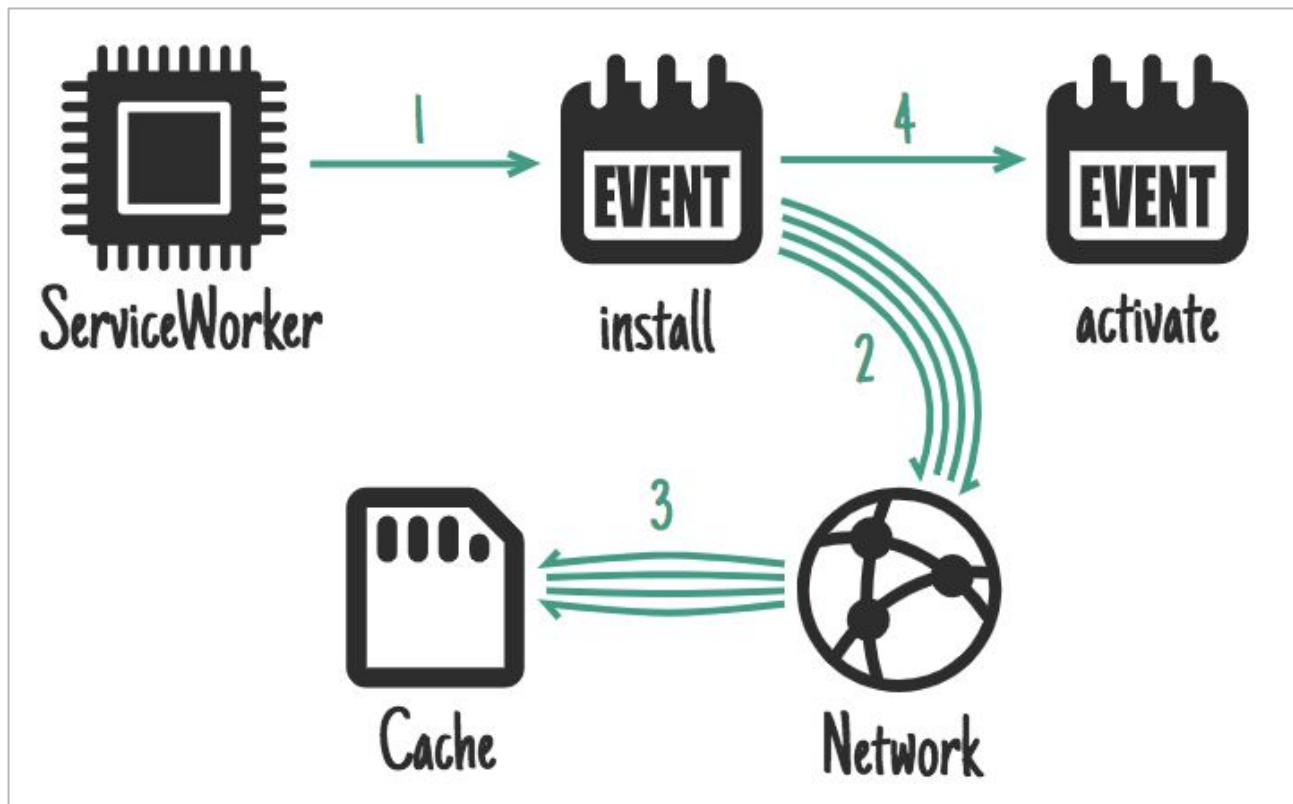
Install workbox-cli



```
npm install workbox-cli
```

Install it locally ↗

Precaching



code



```
module.exports = {  
  
  globDirectory : './', // where workbox should search  
  globPatterns : [ // what workbox should cache  
    'index.html',  
    'css/*.css',  
    'img/*.{png,svg}'  
  ],  
  
  swDest: './sw.js', // the destination of my Service Worker  
  clientsClaim: true,  
  skipWaiting: true  
}
```



*We create a
workbox-config.js
At the root of our project*

Generate the Service Worker



```
npx workbox generateSW
```

*npx to run a binary
in node_modules*

```
importScripts("https://storage.googleapis.com/workbox-cdn/releases/3.6.3/workbox-sw.js");

workbox.skipWaiting();
workbox.clientsClaim();

self.__precacheManifest = [
  {
    "url": "index.html",
    "revision": "d41c1545d09be11d73d187111350da5f"
  },
  {
    "url": "css/app.css",
    "revision": "064247e82330e7818d8c756d1b919b88"
  },
  {
    "url": "img/Bitcoin.svg",
    "revision": "2a4ac9a43a95d361f50010e2f85039b5"
  },
  {
    "url": "img/Dash.svg",
    "revision": "64c25337498c0b8d799c140cb0f8c0b3"
  },
  {
    "url": "img/Ethereum.svg",
    "revision": "dadbeae8504595321d65842442d2f5d1"
  }
].concat(self.__precacheManifest || []);
workbox.precaching.suppressWarnings();
workbox.precaching.precacheAndRoute(self.__precacheManifest, {});
```

←
The generated
service worker

```
(async () => {  
  if ("serviceWorker" in navigator) {  
    // we register our service worker  
    const registration = await navigator.serviceWorker.register('/sw.js');  
    // when our service worker is updated  
    registration.onupdatefound = () => {  
      // when our service worker is updated  
      registration.installing.onstatechange = function () {  
        console.log(`Service worker... ${this.state}`);  
      };  
    };  
  }  
})();  
.catch(e => console.log(`🤖 : ${e}`));
```

← We register our
Service worker

[workbox PWA](#)

Pixel 2 411 x 731 100%

Elements Network Console Sources Performance Memory Application Security Audits

View: Group by frame Preserve log Disable cache Offline Offline

Filter Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other

Name

- localhost
- icon?family=Material+Icons
- materialize.min.css
- materialize.min.js
- app.js
- app.css
- sw.js

Headers Preview Response Timing

General

Request URL: <https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js>

Referrer Policy: no-referrer-when-downgrade

Request Headers

Provisional headers are shown

Referer: http://localhost:4598/

User-Agent: Mozilla/5.0 (Linux; Android 8.0; Pixel 2 Build/OPD3.170816.012) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Mobile Safari/537.36

Not fully working offline...

How to **cache** **external assets** ?



```
https://cdnjs.cloudflare.com/*  
https://fonts.googleapis.com/*
```

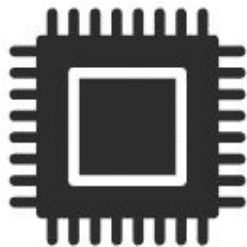

■ Stale-while-revalidate



Page



Cache



ServiceWorker



Network

←
“Ideal for: Frequently
updating resources
where having the very
latest version is
non-essential.”



```
// All the request which will match /(https:\\\\cdnjs.cloudflare.com\\/ajax\\/libs\\/materialize)/ will fall in  
the strategy  
// I also specify that It will be only for GET requests  
workbox.routing.registerRoute(/(https:\\\\cdnjs.cloudflare.com\\/ajax\\/libs\\/materialize)/,  
workbox.strategies.staleWhileRevalidate(), 'GET' );
```

Pixel 2 411 x 731 100%

workbox PWA



Bitcoin

3539 €
2018-12-10T18:00:00Z



Dash

67 €
2018-12-10T18:00:00Z



Ethereum

91 €
2018-12-10T18:00:00Z

Application

- Manifest
- Service Workers
- Clear storage

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies

Cache

- Cache Storage
 - workbox-precache-http://localhost:4598/-ter
 - workbox-precache-http://localhost:4598/ - h
 - workbox-runtime-http://localhost:4598/ - http**
 - Application Cache

Frames

- top

Sources Performance Memory **Application** Security Audits

Path	Content-Type	Content-Length	Time Cache
/ajax/libs/materialize/1.0.0/css/materialize.min.css	text/css	0	10/12/201
/ajax/libs/materialize/1.0.0/js/materialize.min.js	application/jav...	0	10/12/201

A new cache "runtime"
has been created

workbox PWA

Network										
View: [icon] [icon] Group by frame [icon] Preserve log [x] Disable cache [x] Offline Offline [v]										
Filter [x] Hide data URLs [x] All XHR JS CSS Img Media Font Doc WS Manifest Other										
Name	Method	Status	Domain	Type	Initiator	Size	Time	Co...	Waterfall	
localhost	GET	200	localhost	document	Other	(from ServiceW...	14 ms			
icon?family=Material+Icons	GET	(failed)	fonts.googleapi...	stylesheet	(index)	0 B	5 ms			
app.js	GET	(failed)	localhost	script	(index)	0 B	26 ms			
materialize.min.css	GET	200	cdnjs.cloudflare...	stylesheet	(index)	(from ServiceW...	26 ms	br		
app.css	GET	200	localhost	stylesheet	(index)	(from ServiceW...	32 ms			
materialize.min.js	GET	200	cdnjs.cloudflare...	script	(index)	(from ServiceW...	38 ms	br		
materialize.min.css	GET	(failed)	cdnjs.cloudflare...	fetch	work...	0 B	4 ms			
materialize.min.js	GET	(failed)	cdnjs.cloudflare...	fetch	work...	0 B	9 ms			
sw.js	GET	200	localhost	javascript	sw.js	0 B	4 ms			

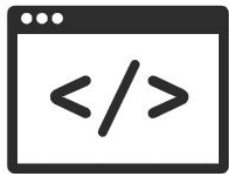
Our AJAX call
are failing...

How to **cache** “API” data ?



`https://api.nomics.com`

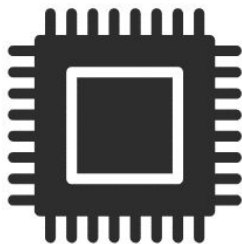
■ Network falling back to cache



Page



Cache



ServiceWorker



Network



*“Ideal for :
requests that are
updating
frequently*



```
// I'm caching all requests which match https://api.nomics.com/*  
workbox.routing.registerRoute(/(https:\/\/api.nomics.com\/)/, workbox.strategies.networkFirst(), 'GET' );
```

*The requests falling in
this strategy*



*The networkFirst
strategies*



GET requests only



workbox PWA



Bitcoin

3539 €

2018-12-10T18:00:00Z



Dash

68 €

2018-12-10T18:00:00Z



ethereum

Ethereum

91 €

2018-12-10T18:00:00Z

Network										
View: [Icons] Group by frame [] Preserve log [x] Disable cache [x] Offline Online [v]										
Filter [] Hide data URLs [x] All XHR JS CSS Img Media Font Doc WS Manifest Other										
Name	Method	Status	Domain	Type	Initiator	Size	Time	Co...	Waterfall	
localhost	GET	200	localhost	document	Other	(from ServiceW...	15 ms			
materialize.min.css	GET	200	cdnjs.cloudflare...	stylesheet	(index)	(from ServiceW...	33 ms	br		
app.css	GET	200	localhost	stylesheet	(index)	(from ServiceW...	48 ms			
materialize.min.js	GET	200	cdnjs.cloudflare...	script	(index)	(from ServiceW...	52 ms	br		
app.js	GET	200	localhost	script	(index)	(from ServiceW...	51 ms			
icon?family=Material+Icons	GET	200	fonts.googleapi...	stylesheet	(index)	1.3 KB	28 ms	br		
materialize.min.css	GET	200	cdnjs.cloudflare...	fetch	work...	20.0 KB	66 ms	br		
materialize.min.js	GET	200	cdnjs.cloudflare...	fetch	work...	40.1 KB	63 ms	br		
interval?key=03a9cb810b88f5519...	GET	200	api.nomics.com	fetch	app.j...	(from ServiceW...	365 ms	gzip		
interval?key=03a9cb810b88f5...	GET	200	api.nomics.com	fetch	work...	819 B	356 ms	gzip		
interval?key=03a9cb810b88f5519...	GET	200	api.nomics.com	fetch	app.j...	(from ServiceW...	122 ms	gzip		
Bitcoin.svg	GET	200	localhost	svg+xml	app.j...	(from ServiceW...	8 ms			
interval?key=03a9cb810b88f5...	GET	200	api.nomics.com	fetch	work...	810 B	111 ms	gzip		
interval?key=03a9cb810b88f5519...	GET	200	api.nomics.com	fetch	app.j...	(from ServiceW...	121 ms	gzip		
Dash.svg	GET	200	localhost	svg+xml	app.j...	(from ServiceW...	9 ms			
interval?key=03a9cb810b88f5...	GET	200	api.nomics.com	fetch	work...	1.0 KB	110 ms	gzip		
Ethereum.svg	GET	200	localhost	svg+xml	app.j...	(from ServiceW...	5 ms			
sw.js	GET	200	localhost	javascript	sw.js	0 B	35 ms			

It WORKS !!



Fork the project on Github

bit.ly/pwa-workbox

PWA

Add workbox to your
build process



workbox-config.js



```
module.exports = {
  globDirectory: './',
  globPatterns: [
    'css/app.css',
    'js/app.bundle.js',
    'img/header-logo-white.png',
    'img/header-logo-small-black.png'
  ],
  swSrc: './serviceworker-dev.js',
  swDest: './serviceworker.js'
};
```

serviceworker-dev.js

```
importScripts("https://storage.googleapis.com/workbox-cdn/releases/3.0.1/workbox-sw.js");

workbox.skipWaiting();
workbox.clientsClaim();

workbox.core.setLogLevel(workbox.core.LOG_LEVELS.debug);

workbox.precaching.precacheAndRoute([]); // workbox-cli will update this with our precache at build time

// cache our offline page
workbox.precaching.precacheAndRoute([
  {
    "url": "/offline/",
    "revision": "d3ab5add5def6bsse1864b1b4719e5e54ff21"
  }
]);

workbox.precaching.suppressWarnings();

workbox.routing.registerRoute(/(https:\\\\fonts.googleapis.com)/, workbox.strategies.staleWhileRevalidate(),
'GET');

workbox.routing.registerRoute('/', (args) => {

  return workbox.strategies.networkFirst()
    .handle(args).then((response) => (!response) ? caches.match('/offline/') : response);

});
```

```
importScripts("https://storage.googleapis.com/workbox-cdn/releases/3.0.1/workbox-sw.js");

workbox.skipWaiting();
workbox.clientsClaim();

workbox.core.setLogLevel(workbox.core.LOG_LEVELS.debug);

/**
 * The workboxSW.precacheAndRoute() method efficiently caches and responds to
 * requests for URLs in the manifest.
 * See https://goo.gl/S9QRab
 */
workbox.precaching.precacheAndRoute([
  {
    "url": "/css/app.css",
    "revision": "3c4e05bd114d1c95512cfe40455d8278"
  },
  {
    "url": "/js/libs/turbolinks.js",
    "revision": "1b00a59fb310465de63767543dc28b28"
  },
  {
    "url": "/js/app.bundle.js",
    "revision": "36ee76ca9774d7fc1b649973c1285215"
  },
  {
    "url": "/img/header-logo-white.png",
    "revision": "294290a2e916c3203f8707d5f4864499"
  },
  {
    "url": "/img/header-logo-small-black.png",
    "revision": "734a0b829f739b42e42df4980e6cd961"
  }
]);

// cache our offline page
// revision should be updated manually here
workbox.precaching.precacheAndRoute([
```

PWA conclusion





UX Workshops

Your password :

ex : 471339609

login

TheUxWorkShops.com
Password : speedRace

antoineBr@

ne-speedrace@google.com



PWA

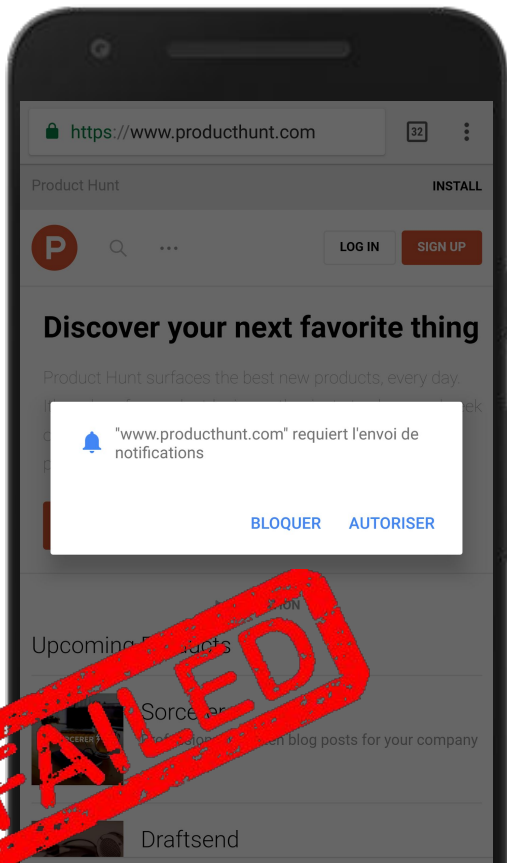
Push Notification







Step 3 : push notifications



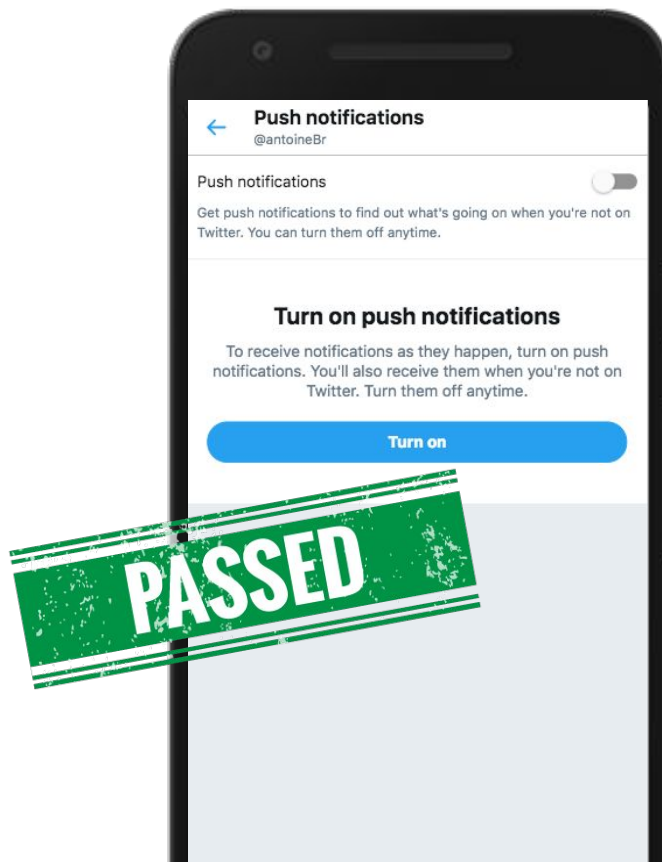
Don't **force** push notification

First think about alternatives !

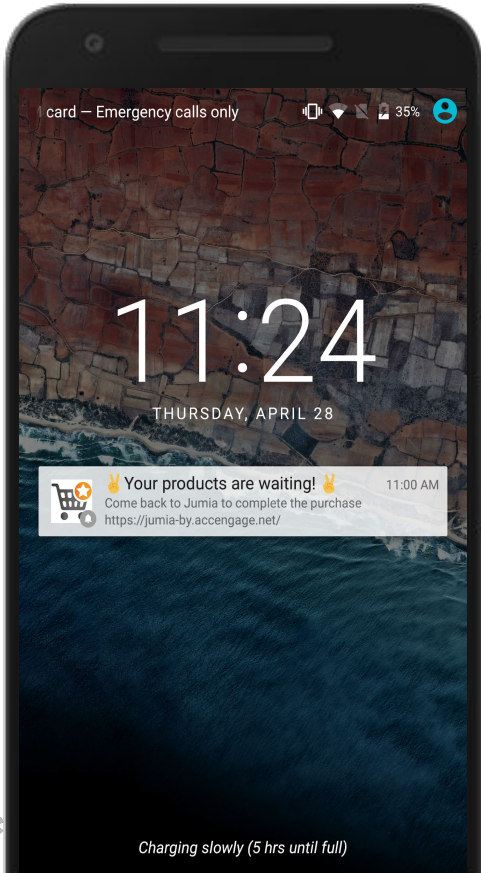
Keep in mind : a good notification is relevant and **contains information that can't be deferred**



Step 3 : push notifications



Step 3 : push notifications



Implementation time : 2 H



1) Choose a platform



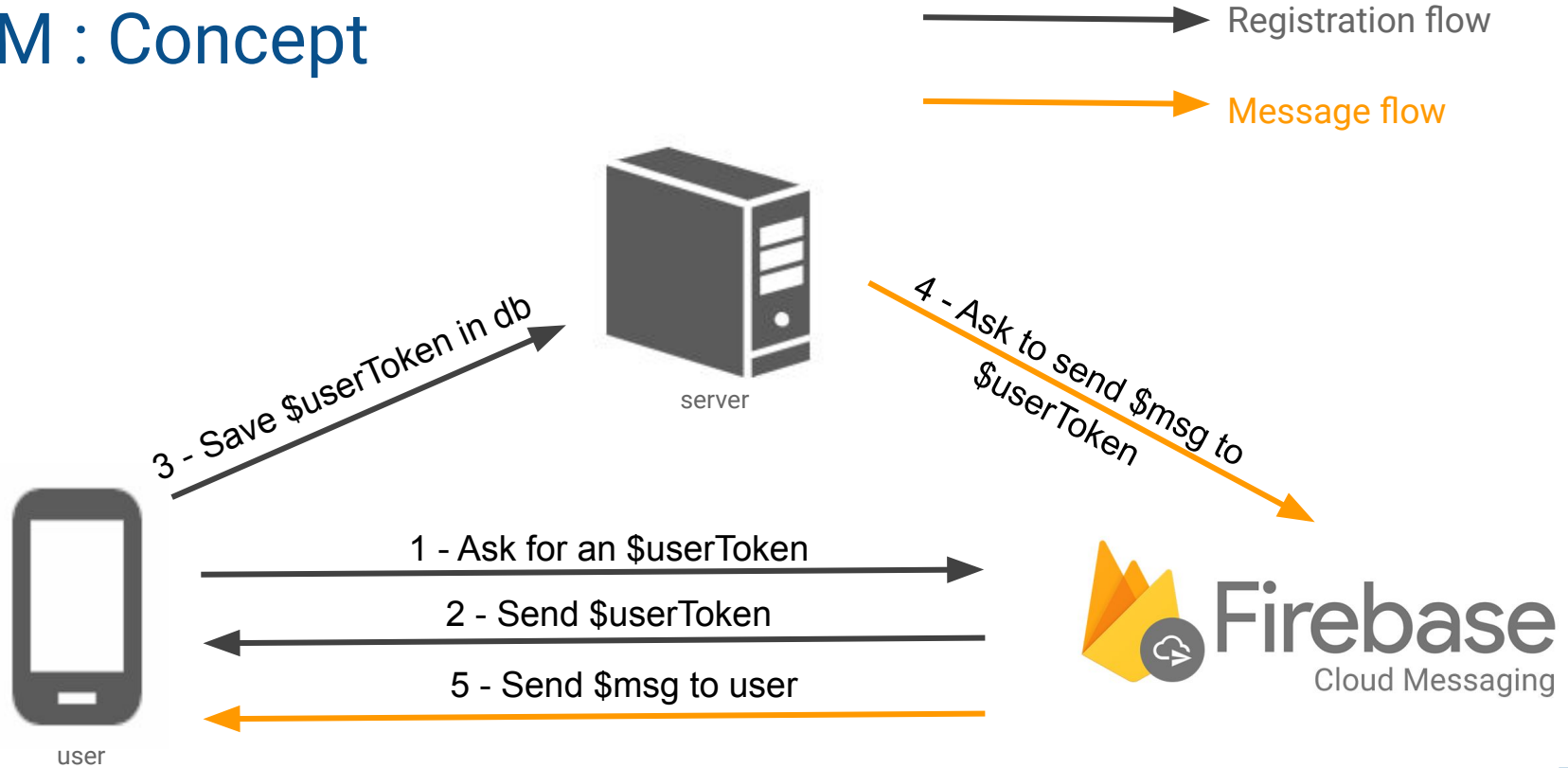
2) Setup the SDK



3) Send notifications (API)



FCM : Concept



Notifications: choose the platform



Notifications : One signal



Easy to implement



A dashboard + REST API



Notifications from your domain



Customizable



Free



Notifications : One signal



Free : But oneSignal **will collect data on your site**

“Web pages visited that have implemented the SDK , and information about those visits (e.g., session duration, time-stamp, referring URLs) ...”



Notifications : Firebase



Need to write server side code



No dashboard but a REST API



Notifications from your domain



Customizable



Free



Notifications : Home Made



Need to write server side code



You have to create your API



Notifications from your domain



Customizable



Free



Notifications : Firebase



Create an account



Add the firebase client library



Initialize Firebase



Request user permission and token



Create service worker



Send notifications



Notifications : One signal

```
<script src="https://cdn.onesignal.com/sdks/OneSignalSDK.js" async=""></script>
<script>
  var OneSignal = window.OneSignal || [];
  OneSignal.push(function() {
    OneSignal.init({
      appId: "26e25b26-86b2-4234-bd44-3b7f9365c735",
      autoRegister: false,
      notifyButton: {
        enable: false,
      },
    });
  });
</script>
```



Notifications : One signal, custo . . .

```
<a href="#" id="subscribe-link" style="display: none;">Subscribe to Notifications</a>

<script>
  function subscribe() {
    OneSignal.push(["registerForPushNotifications"]);
    event.preventDefault();
  }

  var OneSignal = OneSignal || [];
  /* This example assumes you've already initialized OneSignal */
  OneSignal.push(function() {
    // If we're on an unsupported browser, do nothing
    if (!OneSignal.isPushNotificationsSupported()) return;
    OneSignal.isPushNotificationsEnabled(function(isEnabled) {

      if (isEnabled) {
        // The user is subscribed to notifications
        // Don't show anything
      } else {
        document.getElementById("subscribe-link").addEventListener('click', subscribe);
        document.getElementById("subscribe-link").style.display = '';
      }
    });
  });
</script>
```



Notifications : One signal

OneSignal

SETTINGS

MESSAGES

USERS

DELIVERY

TEST APP

?

New Message

MessagesA/B TestsTemplatesAutomated

1 Audience

☒ Send to Everyone

☐ Send to Particular Segment(s)

☐ Send to Test Device(s)

2 Message

ENGLISH

TITLE

Hello PWA WorkShop 🦊

MESSAGE

I hope you enjoy this presentation

11:17

🔍

⋮

 Hello PWA WorkShop 🦊
spa.antoinebrossault.com
I hope you enjoy this presentation



Notifications : One signal, subscribe to a topic



```
OneSignal.sendTag("weather", "paris", function(tagsSent) {  
    // Callback called when tags have finished sending  
});
```



Notifications : One signal, create a segment

Segment Editor

 Unnamed Segment 

≈ 1

Users in this segment must meet these rules:

 USER TAG

is

▼



ADD FILTER

ADD OR

≈ 1



Wrap-up

PWA :



The web is back



No reason to not implement the basics of PWAs



Forget 90% of native apps



Title here

```
console.log( “Thanks” );
```

