

# Service Worker

“Man-in-the-middle API for your web apps”





- Nah, Service Workers are not intended to hurt you
- Opposite of that, they are intended to provide you, as a user, with some decent experience once your internet connections goes haywire



- Nah, those are not service workers.
- Those are just consumers that after long day of working sometimes expects your web application to be functioning properly.



## There is no Internet connection

Try:

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

ERR\_INTERNET\_DISCONNECTED

Go from shitty experience such as this



- To having offline experience like this
- Where content is still available
- There is an actual working example where I added small service worker to Vilnius.js website code to cache some of the resources, so you can still visit the website as if it was some app once you go offline

Elements

Console

Sources

Network

Performance

Memory

>>

7

3

View:

Preserve log

Disable cache

Offline

Offline (0ms, 0kb/s, 0kb

Filter

Regex

Hide data URLs

All

XHR

JS

CSS

Img

Media

Font

Doc

WS

Manifest

Other

Name	Status	Type	Initiator	Size	Time	Waterfall
localhost	200	document	Other	(from ServiceWorker)	16...	
foundation.min.css	200	stylesheet	(index)	(from ServiceWorker)	7ms	
style.css	200	stylesheet	(index)	(from ServiceWorker)	7ms	
rotem_mizrachi-meidan.png	200	png	(index)	(from ServiceWorker)	8ms	
peter_muller.png	200	png	(index)	(from ServiceWorker)	39...	
track.js	200	script	(inde...	(from ServiceWorker)	10...	
track.js	200	x-javascript	Other	(from disk cache)	5ms	
logo.svg	200	svg+xml	onloa...	(from ServiceWorker)	4ms	
github.svg	200	svg+xml	onloa...	(from ServiceWorker)	6ms	
twitter.svg	200	svg+xml	onloa...	(from ServiceWorker)	6ms	
email.svg	200	svg+xml	onloa...	(from ServiceWorker)	7ms	
rss.svg	200	svg+xml	onloa...	(from ServiceWorker)	6ms	
wix.jpg	200	jpeg	onloa...	(from ServiceWorker)	4ms	
mobilelife.jpg	200	jpeg	onloa...	(from ServiceWorker)	4ms	

“Script that acts as **proxy** server that sits between your **web app**, **browser** and the **network**.”

- The definition of service workers, sometimes they are confused with Web Workers
- “Web Workers are a mechanism by which a script operation can be made to run in a background thread separate from the main execution thread of a web application.”

# Provide user with

- Consistent performance
- Native like experience
- Reduced data usage

- Reliable performance that is consistently fast (*Repeat visits are extremely quick. Static assets and the UI (e.g. HTML, JavaScript, images and CSS) are cached on the first visit so that they load instantly on repeat visits. Content may be cached on the first visit, but is typically loaded when it is needed.*)
- Native-like interactions (*By adopting the **app shell model**, you can create experiences with instant, native-application-like navigation and interactions, complete with offline support.*)
- Economical use of data (*Design for minimal data usage and be judicious in what you cache because listing files that are non-essential (large images that are not shown on every page, for instance) result in browsers downloading more data than is strictly necessary. Even though data is relatively cheap in western countries, this is not the case in emerging markets where connectivity is expensive and data is costly.*)



# Lifecycle

# Stages

**1. Download**

**2. Install**

**3. Activate**

# Installing

Installing

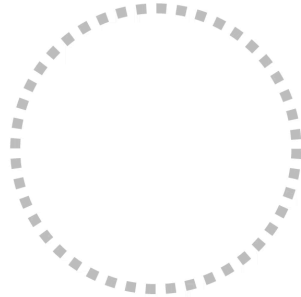


Active

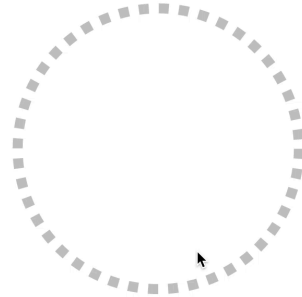


# Updating

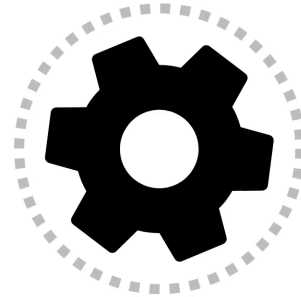
Installing



Waiting



Active














**Internals**

- **No DOM access**
- **Can run without page**
- **Terminates and runs itself**
- **HTTPS only**

- Browser spins yet another instance of JS engine in order to execute your scripts, so all the execution happens in another thread
- Works for localhost on http, in order to do some testing without going through the clutter of setting up SSL

# Scopes

Application	
 Manifest	
 Service Workers	
 Clear storage	
Storage	
▶  Local Storage	
▶  Session Storage	
▶  IndexedDB	
 Web SQL	
▶  Cookies	
Cache	
▶  Cache Storage	
 Application Cache	
Frames	
▶  top	

<b>https://googlechrome.github.io/samples/service-worker/post-message/</b>	<a href="#">Update</a>
Source <a href="#">service-worker.js</a> Received 1970-1-1 03:00:00	
Status <span>●</span> #3703 activated and is stopped <a href="#">start</a>	

**https://www.smashingmagazine.com/**	[Update](#)
Source [serviceWorker.js](#) Received 2017-4-22 20:30:53	
Status ● #4700 activated and is stopped [start](#)	
Clients [https://www.smashingmagazine.com/2016/02/making-a-service-worker/](#) [focus](#)	
**https://drive.google.com/drive**	[Update](#)
Source [serviceworker.js?ouid=udeadff6dc1bb88ed](#) Received 2017-4-23 21:49:49	
Status ● #4939 activated and is running [stop](#) [inspect](#)	
Clients [https://drive.google.com/drive/folders/0B8wIHxDgKhseRHc4a0NsUkFhYTQ](#) [focus](#)	

# Related APIs

- Service Workers left alone is not that much of a useful thing
- It relies on many other APIs such as the following
- **NEXT SLIDE:** DEPENDENT APIs



Promise  
Fetch  
Broadcast Channel  
Cache  
IndexedDB

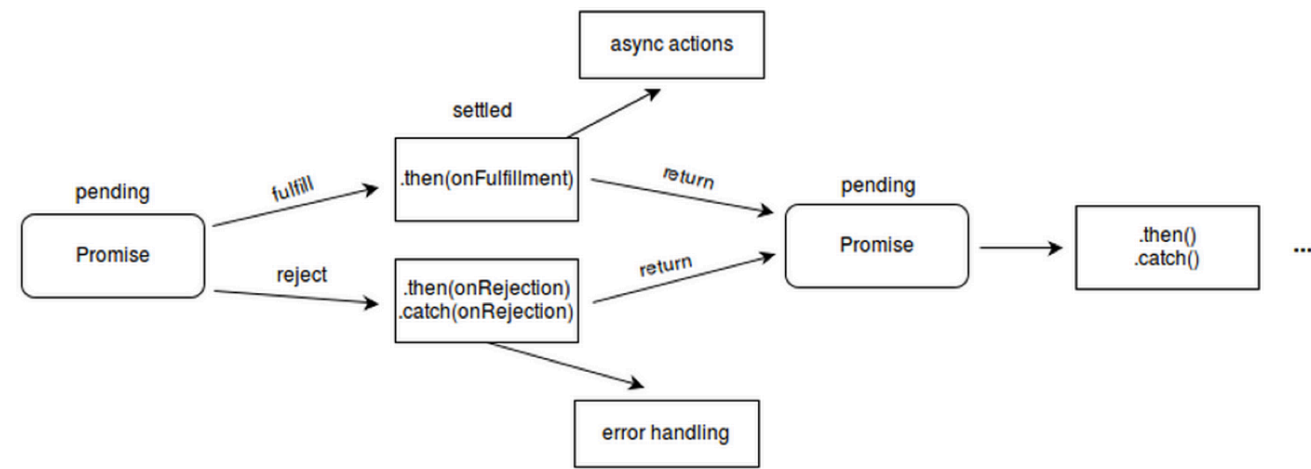
- More on Cache: <https://developer.mozilla.org/en-US/docs/Web/API/Cache>

**PROVIDE BRIEF INTRO TO MOST OF THEM**

EVEN THOUGH IT'S NOT ENTIRELY NECESSARY TO HAVE A GRASP OF ANYTHING ELSE THAN Promises and Fetch API

# Promises

- **Value** which may be available **now** in the **future** or **never**

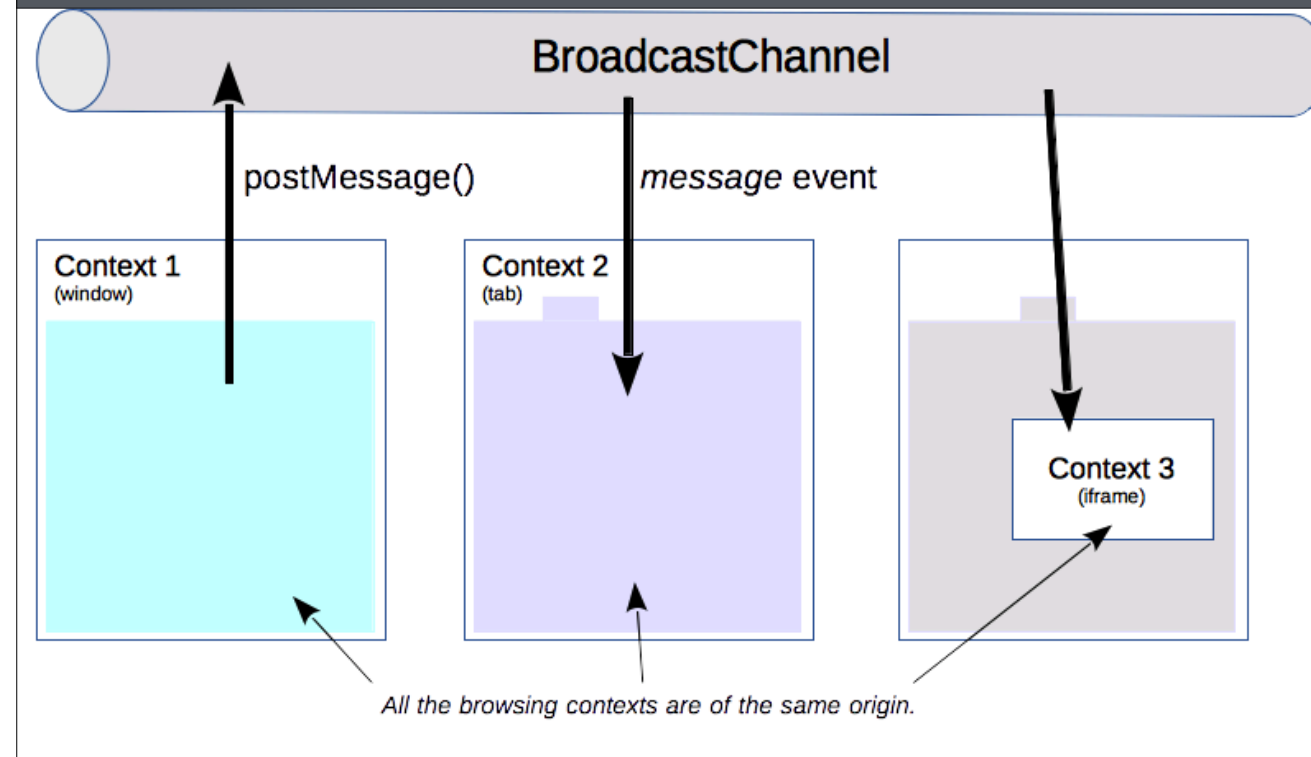


# Fetch API

- Replaces **XMLHttpRequest**
- Uses **Promises**
- Generic definitions for:
  - **Request/Response**
  - **Headers**

Fetch provides a generic definition of Request and Response objects (and other things involved with network requests). This will allow them to be used wherever they are needed in the future, whether it's for service workers, Cache API and other similar things that handle or modify requests and responses, or any kind of use case that might require you to generate your own responses programmatically.

# Broadcast Channel



# Cache

Application			
<div>Manifest</div> <div>Service Workers</div> <div>Clear storage</div>			
<div>Storage</div> <div>Local Storage</div> <div>Session Storage</div> <div>IndexedDB</div> <div>Web SQL</div> <div>Cookies</div>			
<div>Cache</div> <div>Cache Storage</div> <div>newtab-static-97ulrx - https://</div> <div>Application Cache</div>			
<div>Frames</div> <div>top</div>			
#	Request	Response	
0	https://apis.google.com/_scs/abc-static/_js/k=gapi.gapl.en.DTPeBB_SvOA.O/m=gapi_iframes,goo...		
1	https://lh3.googleusercontent.com/--U8ZtjEryrw/AAAAAAAAAI/AAAAAAAAAA/HAhGhrQOtyscMn...		
2	https://ssl.gstatic.com/chrome/components/doodle-notifier-01.html		
3	https://ssl.gstatic.com/chrome/components/doodle-notifier-02.html		
4	https://ssl.gstatic.com/gb/images/v1_b3735dd8.png		
5	https://www.google.it/_chrome/newtab?espv=2&ie=UTF-8		
6	https://www.google.it/images/branding/googlelogo/2x/googlelogo_color_272x92dp.png		
7	https://www.google.it/images/branding/googlelogo/2x/googlelogo_light_color_272x92dp.png		
8	https://www.google.it/images/srpr/chrome_ntp_white_logo2.png		
9	https://www.google.it/images/srpr/logo11w.png		
10	https://www.google.it/xjs/_js/k=xjs.ntp.en_US.DmkNsH-BVO8.O/m=jsa,ntp,d,csi/am=BCI/rt=j/d=1/t...		
11	https://www.google.it/xjs/_js/k=xjs.ntp.en_US.DmkNsH-BVO8.O/m=spch/am=BCI/rt=j/d=0/t=zcms/...		
12	https://www.gstatic.com/og/_js/k=og.og.en_US.o-SyWnQPMWo.O/rt=j/m=id,gl,is,id,nb,nw,sb,sd,p,v...		

# IndexedDB

Application			
Start from key			
Application			
Manifest			
Service Workers			
Clear storage			
Storage			
Local Storage			
http://localhost:8888			
Session Storage			
IndexedDB			
witr - http://localhost:8888			
witr			
by-date			
Web SQL			
Cookies			
Cache			
Cache Storage			
Application Cache			
Frames			
top			
#	Key (Key path: "id")	Value	
0	"10u8y5sz2f5"	Object	
		avatar: "/avatars/lillie"	
		body: "takes a lot the level one heads around. It stuff"	
		id: "10u8y5sz2f5"	
		name: "Lillie Wolfe"	
		photo: "/photos/5-4001-4285585982-5244f4168c"	
		time: "2017-04-24T05:34:16.660Z"	
1	"2p75ha92bcs"	Object	
2	"4efyhu5f8qm"	Object	
3	"55bnuiuwu80"	Object	
4	"599g7zj7uvc"	Object	
5	"5im73ymy0w8"	Object	
6	"6ysw2cl00e4"	Object	
7	"92puvgap6go"	Object	
8	"9t3u6gf5ed8"	Object	
9	"b7104avmneo"	Object	
10	"bu2gpgmmayw"	Object	
11	"chlncaync3k"	Object	
12	"df2uiqp8zk0"	Object	
13	"dmgp7zgt9c"	Object	
14	"dxam90bny14"	Object	
15	"f2qxs4ophc"	Object	
16	"g3vai65fk"	Object	
17	"gdm7u4d4mbs"	Object	
18	"ic7rqnd7pq8"	Object	
19	"k4pjv3x84po"	Object	
20	"md213cw9p1c"	Object	
21	"mpk7haxu9z4"	Object	

# Code Example

- Show very basic code example
- Registered, working, caching, working offline



# Browser Support

is  
SERVICEWORKER  
ready?

Status \* Spec \* Intro \* Resources \* GitHub

## ServiceWorker enthusiasm

The first thing any implementation needs.



**Chrome:** Shipped.

**Firefox:** Shipped.

**Samsung Internet:** Shipped. Based on Chromium 44.2403 with some [additions and changes](#). (See "Service Worker" section.)

**Safari:** [Under consideration](#), Brief positive signals in [five year plan](#).

**Edge:** [In development](#).

Support does not include iOS versions of third-party browsers on that platform (see Safari support).

- So yeah, Apple are being an asshole, because why not

GAME OVER



There is no Internet connection

Try:

- Using **Service Workers**

ERR\_INTERNET\_DISCONNECTED

# References

- **Udacity - Intro to Progressive Web Apps**
- **MDN - Service Workers**
- **Web Fundamentals - Service Workers**
- **Offline First**



- All of these are keywords for one to put into Google or some other Search Engine of choice