FH JOANNEUM
University of Applied Sciences

Progressive Web Apps

“These apps aren’t packaged and deployed through stores, they are just websites that took the right vitamins"
(Alex Russell, Developer on the Google Chrome team)

Mobile Apps / Offline Apps

John Feiner

PWA — Like an App

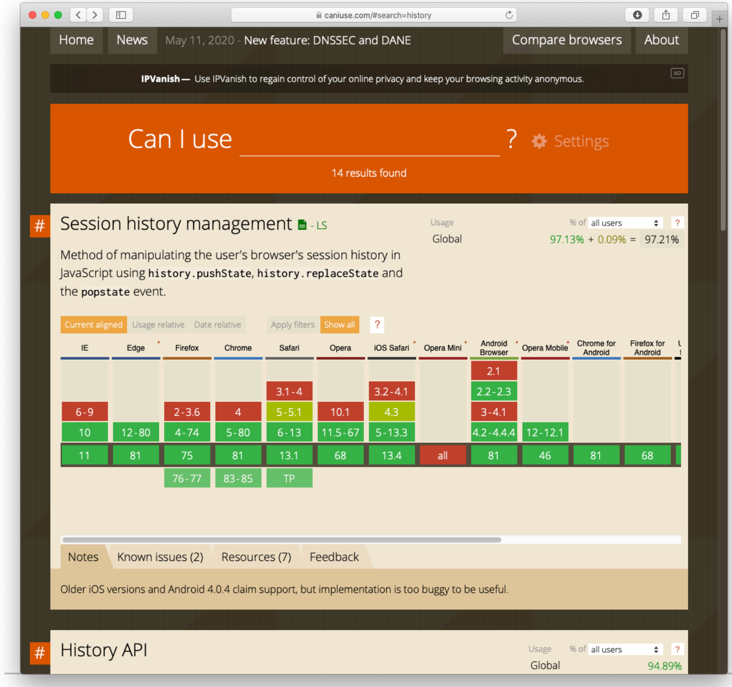
Mobile

+ Install on Desktop
+ Web-App Manifest
+ HomeScreen Button
+ Working Offline
    + Service Workers
    (= caching, network proxy)

See also Sensors, Notifications

https://www.mirror.co.uk/3am/celebrity-news/david-beckham-texts-while-driving-his-motorbike-1391450

What's Available?



caniuse.com/#search=history

Home | News | May 11, 2020 - New feature: DNSSEC and DANE | Compare browsers | About

IPVanish — Use IPVanish to regain control of your online privacy and keep your browsing activity anonymous.

Can I use ___ ? Settings

14 results found

Session history management - LS

Usage: Global — % of all users: 97.13% + 0.09% = 97.21%

Method of manipulating the user's browser's session history in JavaScript using `history.pushState`, `history.replaceState` and the `popstate` event.

Current aligned | Usage relative | Date relative | Apply filters | Show all | ?

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android
								2.1			
				3.1 - 4		3.2 - 4.1		2.2 - 2.3			
6 - 9		2 - 3.6	4	5 - 5.1	10.1	4.3		3 - 4.1			
10	12 - 80	4 - 74	5 - 80	6 - 13	11.5 - 67	5 - 13.3		4.2 - 4.4.4	12 - 12.1		
11	81	75	81	13.1	68	13.4	all	81	46	81	68
		76 - 77	83 - 85	TP							

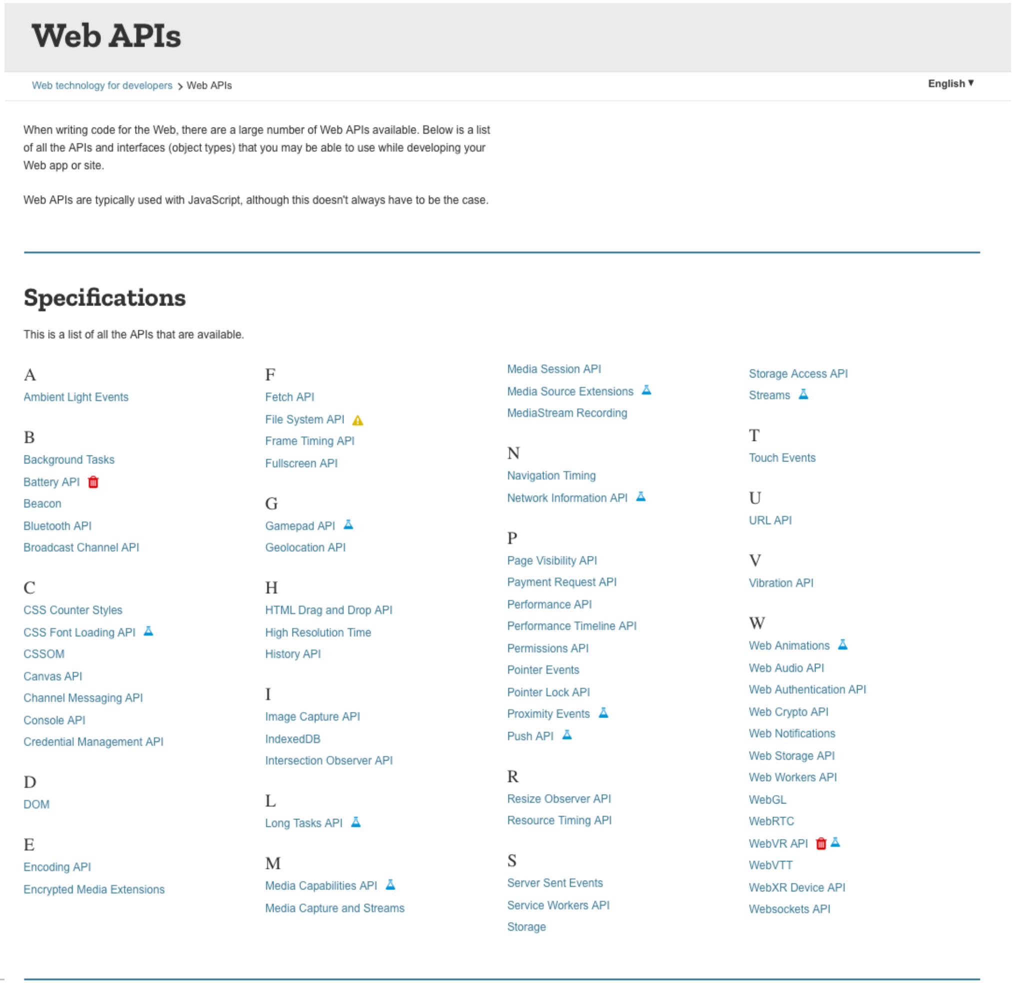
Notes | Known issues (2) | Resources (7) | Feedback

Older iOS versions and Android 4.0.4 claim support, but implementation is too buggy to be useful.

History API

Usage: Global — % of all users: 94.89%

https://caniuse.com



Web APIs

Web technology for developers > Web APIs — English ▾

When writing code for the Web, there are a large number of Web APIs available. Below is a list of all the APIs and interfaces (object types) that you may be able to use while developing your Web app or site.

Web APIs are typically used with JavaScript, although this doesn't always have to be the case.

Specifications

This is a list of all the APIs that are available.

A
- Ambient Light Events

B
- Background Tasks
- Battery API
- Beacon
- Bluetooth API
- Broadcast Channel API

C
- CSS Counter Styles
- CSS Font Loading API
- CSSOM
- Canvas API
- Channel Messaging API
- Console API
- Credential Management API

D
- DOM

E
- Encoding API
- Encrypted Media Extensions

F
- Fetch API
- File System API
- Frame Timing API
- Fullscreen API

G
- Gamepad API
- Geolocation API

H
- HTML Drag and Drop API
- High Resolution Time
- History API

I
- Image Capture API
- IndexedDB
- Intersection Observer API

L
- Long Tasks API

M
- Media Capabilities API
- Media Capture and Streams
- Media Session API
- Media Source Extensions
- MediaStream Recording

N
- Navigation Timing
- Network Information API

P
- Page Visibility API
- Payment Request API
- Performance API
- Performance Timeline API
- Permissions API
- Pointer Events
- Pointer Lock API
- Proximity Events
- Push API

R
- Resize Observer API
- Resource Timing API

S
- Server Sent Events
- Service Workers API
- Storage
- Storage Access API
- Streams

T
- Touch Events

U
- URL API

V
- Vibration API

W
- Web Animations
- Web Audio API
- Web Authentication API
- Web Crypto API
- Web Notifications
- Web Storage API
- Web Workers API
- WebGL
- WebRTC
- WebVR API
- WebVTT
- WebXR Device API
- Websockets API

https://developer.mozilla.org/en-US/docs/Web/API

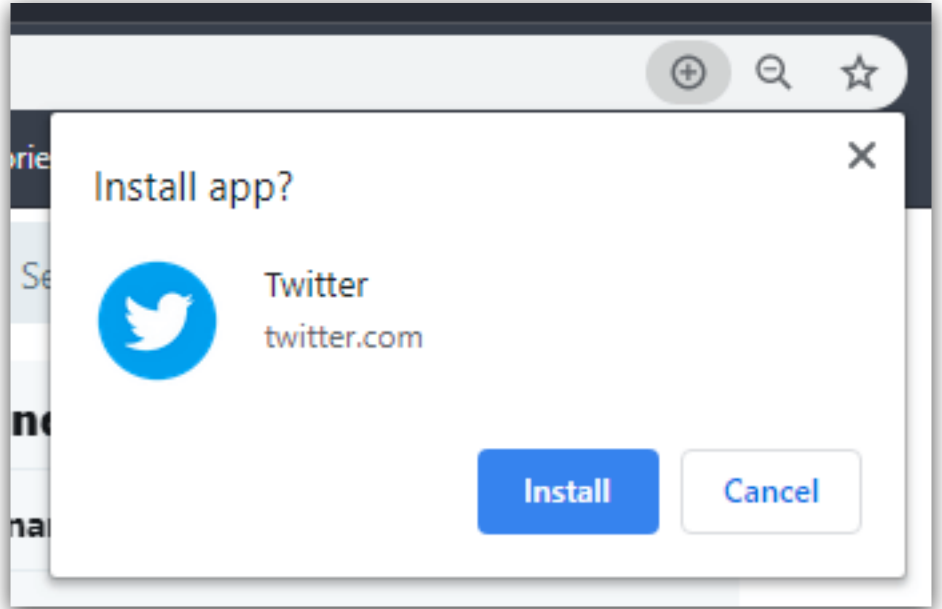
Install on Desktop

Fast launch

Desktop Icon

For

Desktop / iOS / Android


Install app?

Twitter

twitter.com

https://www.simicart.com/blog/desktop-pwa/

Demo Configuration for Multiple Platforms

```
...
 <!-- Add to homescreen for Chrome on Android -->
  <meta name="mobile-web-app-capable" content="yes">
  <link rel="icon" sizes="192x192" href="../../images/touch/chrome-touch-icon-192x192.png">

  <!-- Add to homescreen for Safari on iOS -->
  <meta name="apple-mobile-web-app-title" content="Service Worker Sample: Pre-fetching Resources During Registration">

  <meta name="apple-mobile-web-app-capable" content="yes">
  <meta name="apple-mobile-web-app-status-bar-style" content="black">
  <link rel="apple-touch-icon-precomposed" href="../../images/apple-touch-icon-precomposed.png">

  <!-- Tile icon for Win8 (144x144 + tile color) -->
  <meta name="msapplication-TileImage" content="images/touch/ms-touch-icon-144x144-precomposed.png">
  <meta name="msapplication-TileColor" content="#3372DF">

  <link rel="icon" href="../../images/favicon.ico">
...
```

https://googlechrome.github.io/samples/service-worker/prefetch/

Desktop/iOS/Android

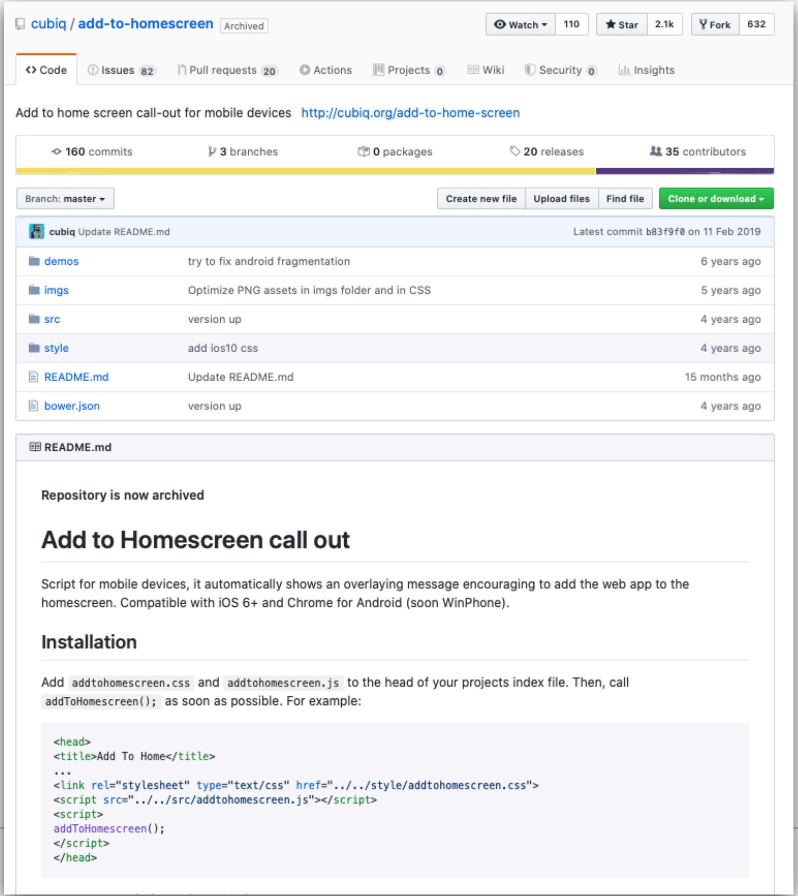
Use helper

```
addToHomescreen();
```

https://github.com/cubiq/add-to-homescreen

iOS: https://developer.apple.com/library/archive/documentation/AppleApplications/Reference/SafariWebContent/ConfiguringWebApplications/ConfiguringWebApplications.html

Android https://codelabs.developers.google.com/codelabs/add-to-home-screen/#0



Add to home screen call-out for mobile devices http://cubiq.org/add-to-home-screen

160 commits | 3 branches | 0 packages | 20 releases | 35 contributors

Branch: master | Create new file | Upload files | Find file | Clone or download

cubiq Update README.md — Latest commit b83f9f0 on 11 Feb 2019

demos	try to fix android fragmentation	6 years ago
imgs	Optimize PNG assets in imgs folder and in CSS	5 years ago
src	version up	4 years ago
style	add ios10 css	4 years ago
README.md	Update README.md	15 months ago
bower.json	version up	4 years ago

README.md

Repository is now archived

Add to Homescreen call out

Script for mobile devices, it automatically shows an overlaying message encouraging to add the web app to the homescreen. Compatible with iOS 6+ and Chrome for Android (soon WinPhone).

Installation

Add `addtohomescreen.css` and `addtohomescreen.js` to the head of your projects index file. Then, call `addToHomescreen();` as soon as possible. For example:

```
<head>
<title>Add To Home</title>
...
<link rel="stylesheet" type="text/css" href="../../style/addtohomescreen.css">
<script src="../../src/addtohomescreen.js"></script>
<script>
addToHomescreen();
</script>
</head>
```

Web App Manifest

whatever.manifest

or manifest.json

```
<!-- Startup configuration -->
<link rel="manifest" href="manifest.webmanifest">
```

```
{
  "short_name": "",
  "name": "",
  "icons": [
    {
      "src":"",
      "sizes": "",
      "type": ""
    }
  ],
  "start_url": "",
  "background_color": "",
  "Theme_color": "",
  "display": ""
}
```

manifest.json

```
Content-Type: application/manifest+json)
```

https://codelabs.developers.google.com/codelabs/add-to-home-screen/#2

https://w3c.github.io/manifest/

FH JOANNEUM
University of Applied Sciences

Service Workers

Minimal latency with caching and prefetching using service worker proxy

John Feiner

How (and why) to Avoid Latency?

Why?

Mobile Scenario: fast startup, save state, short use time, offline usage, …

How?

Caching, preloading, …

Review: Web Worker

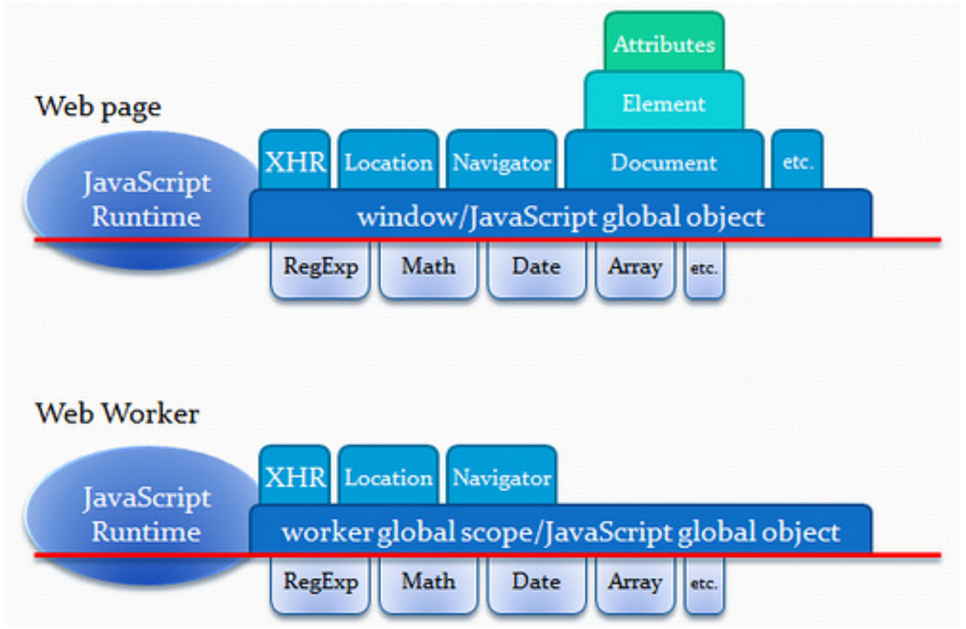
General for concurrency:

Tasks v.s Threads vs. Co-Routines, …

Web Worker:

Dedicated vs. Shared Workers

Text messages (de)serialisation for communication between workers and web page.



Service Worker

Online/Offline:

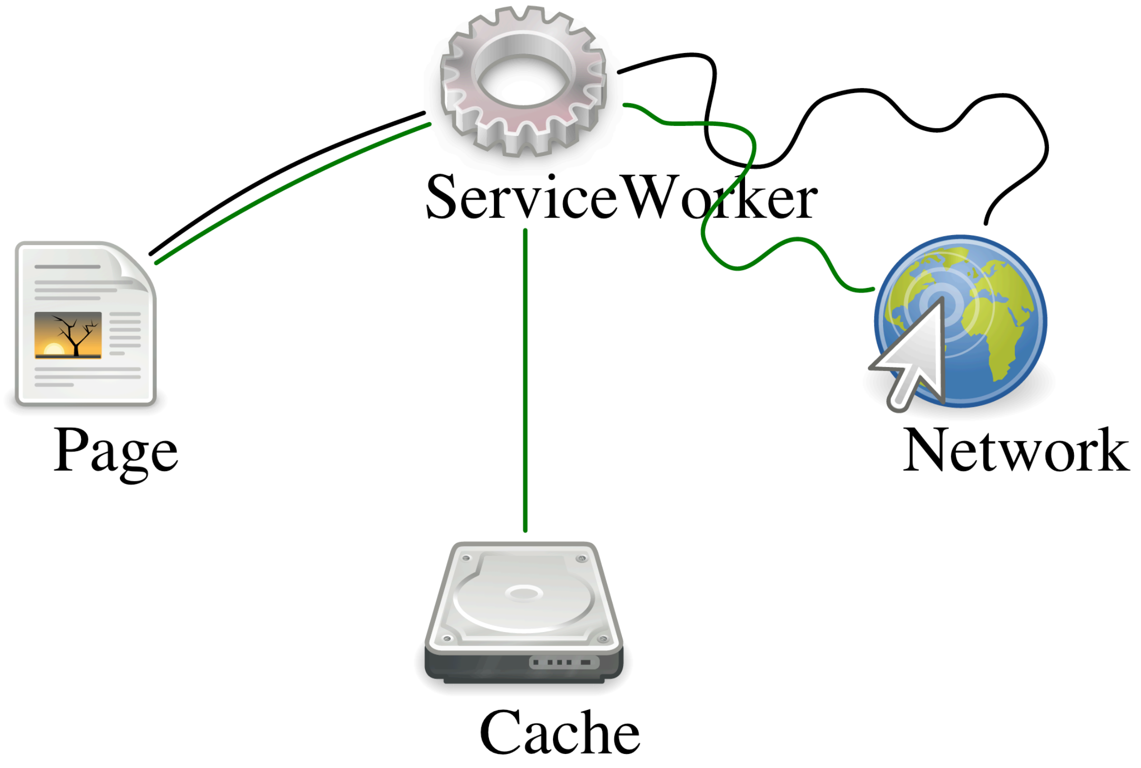
Proxy for caching, fallback, ...

(Web)Push Notification

Background Sync:

defer until net available

Caching

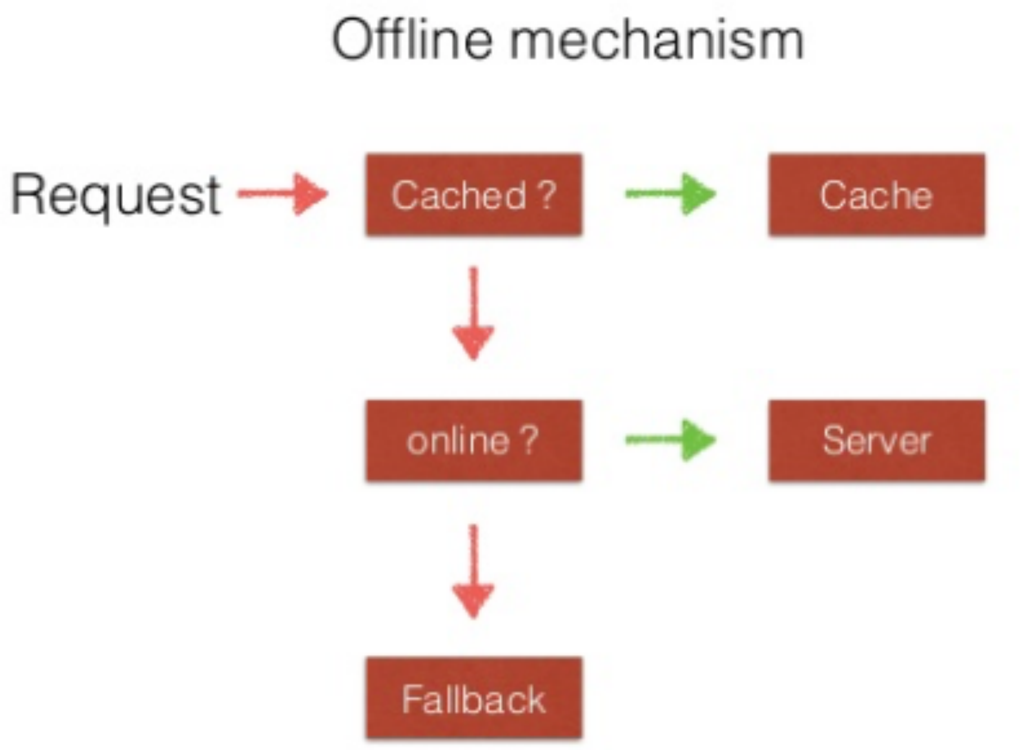


https://infrequently.org/14/html5devconf_oct/#1

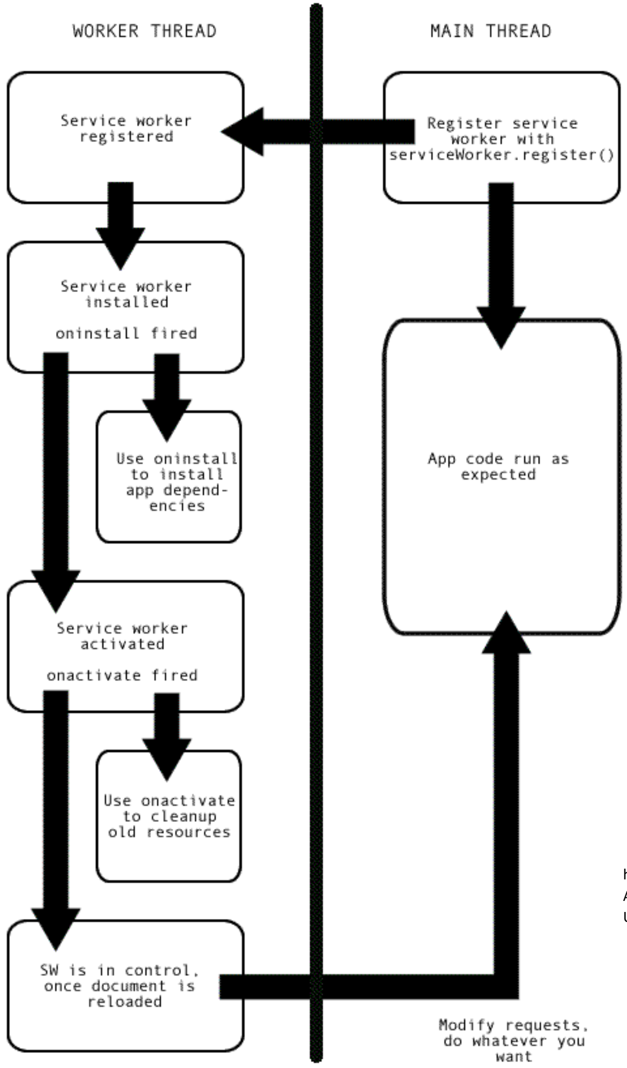
Service Worker

Online/Offline:

proxy for caching, fallback, …



Life-Cycle in Detail


WORKER THREAD
MAIN THREAD
Service worker registered
Register service worker with serviceWorker.register()
Service worker installed
oninstall fired
Use oninstall to install app depend-encies
App code run as expected
Service worker activated
onactivate fired
Use onactivate to cleanup old resources
SW is in control, once document is reloaded
Modify requests, do whatever you want

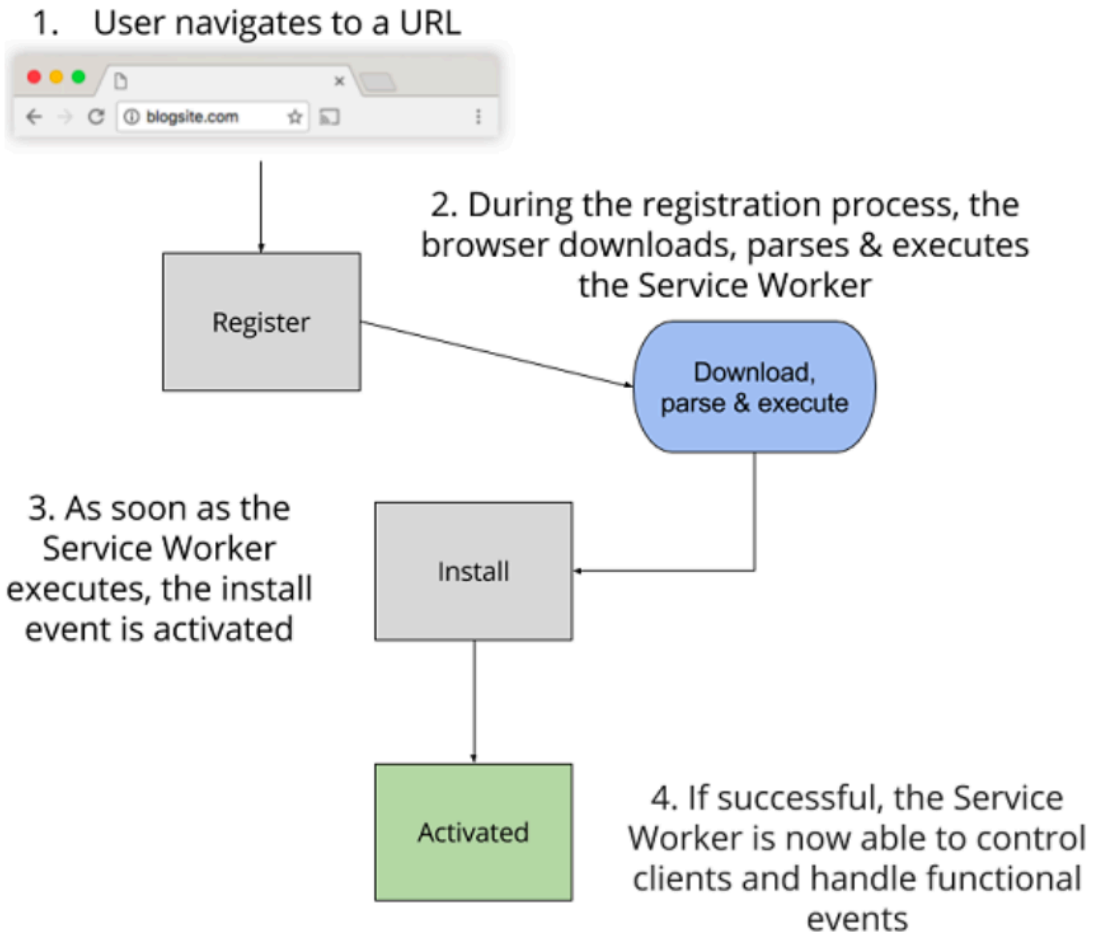
https://developer.mozilla.org/de/docs/Web/API/Service_Worker_API/Using_Service_Workers

Service Worker

Register

Install

Activate


1. User navigates to a URL
blogsite.com
2. During the registration process, the browser downloads, parses & executes the Service Worker
Register
Download, parse & execute
3. As soon as the Service Worker executes, the install event is activated
Install
Activated
4. If successful, the Service Worker is now able to control clients and handle functional events

Service Workers - Code Snippets

```
if ('serviceWorker' in navigator) {
    window.addEventListener('load', function() {
            navigator.serviceWorker.register('01_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);
  });
 });
} else {
    console.log('ServiceWorker not supported');
}
```

Service Workers - Code Snippets

```
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 Workers - Code Snippets

```
self.addEventListener('fetch', function(event) {
  if (/\.jpg$/.test(event.request.url)) {
    event.respondWith(
    fetch('/images/unicorn.jpg, {
       mode: 'no-cors' })
    );
  }
});
```

Fetch API (and Cache and Service Workers)

Fetch returns a **Promise** object:

```
fetch('http://example.com/movies.json')
  .then(function(response) {
    return response.json();
  })
  .then(function(myJson) {
    console.log(JSON.stringify(myJson));
  });
```

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch

Cache API

Usage:

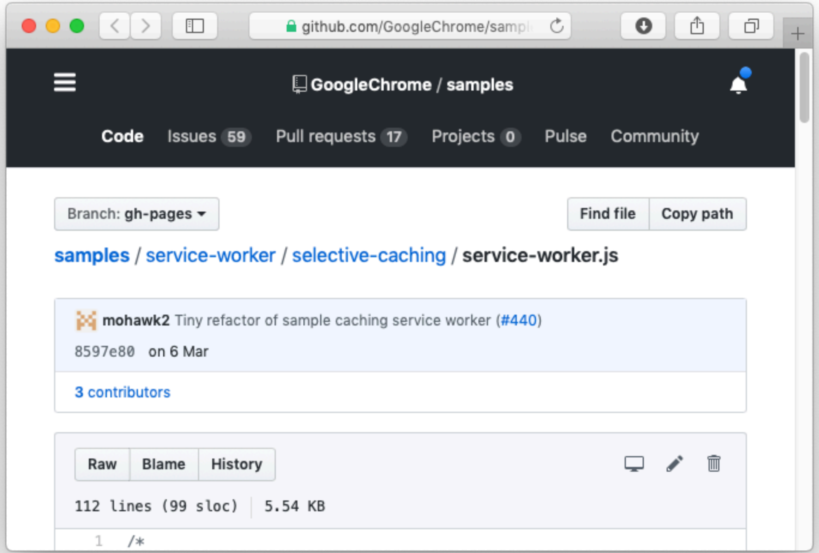
```
cache.put(...)
cache.match(...)
...
```

Fetch and put:

```
cache.add(...)
```

+ Combine with ServiceWorker

- Limited Browser Support



samples / service-worker / selective-caching / service-worker.js

mohawk2 Tiny refactor of sample caching service worker (#440)

8597e80 on 6 Mar

3 contributors

112 lines (99 sloc) | 5.54 KB

```
/*
```

https://github.com/GoogleChrome/samples/blob/gh-pages/service-worker/selective-caching/service-worker.js

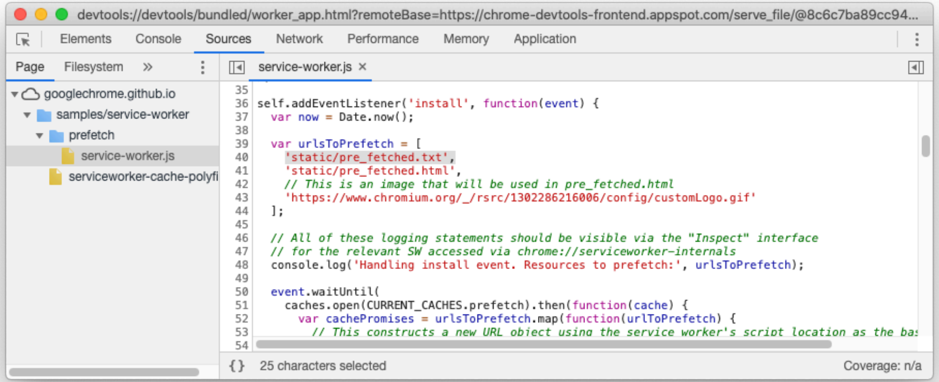
https://developer.mozilla.org/en-US/docs/Web/API/Cache#Browser_compatibility

FH JOANNEUM University of Applied Sciences

Prefetching

During installation phase

chrome://inspect/#service-workers

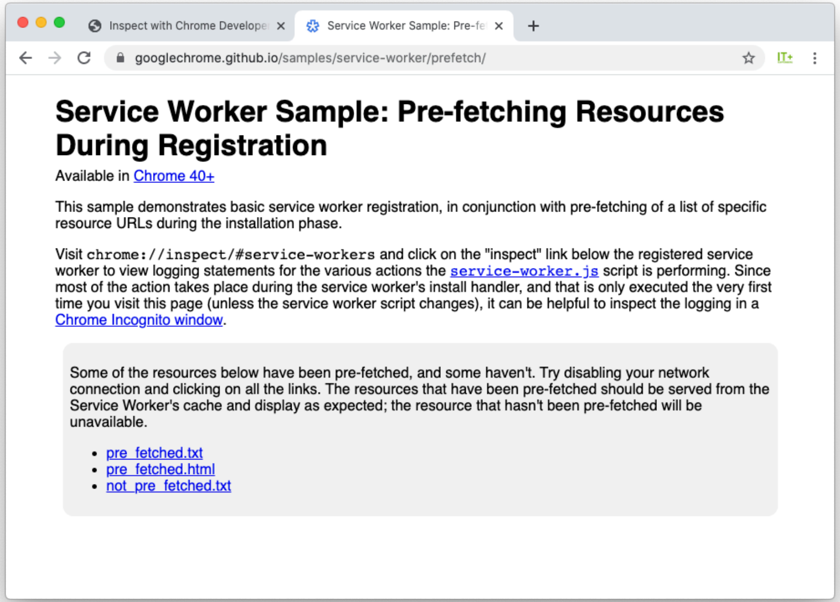


```
self.addEventListener('install', function(event) {
  var now = Date.now();

  var urlsToPrefetch = [
    'static/pre_fetched.txt',
    'static/pre_fetched.html',
    // This is an image that will be used in pre_fetched.html
    'https://www.chromium.org/_/rsrc/1302286216006/config/customLogo.gif'
  ];

  // All of these logging statements should be visible via the "Inspect" interface
  // for the relevant SW accessed via chrome://serviceworker-internals
  console.log('Handling install event. Resources to prefetch:', urlsToPrefetch);

  event.waitUntil(
    caches.open(CURRENT_CACHES.prefetch).then(function(cache) {
      var cachePromises = urlsToPrefetch.map(function(urlToPrefetch) {
        // This constructs a new URL object using the service worker's script location as the bas
```


Inspect with Chrome Develope
Service Worker Sample: Pre-fe
googlechrome.github.io/samples/service-worker/prefetch/

Service Worker Sample: Pre-fetching Resources During Registration

Available in Chrome 40+

This sample demonstrates basic service worker registration, in conjunction with pre-fetching of a list of specific resource URLs during the installation phase.

Visit `chrome://inspect/#service-workers` and click on the "inspect" link below the registered service worker to view logging statements for the various actions the `service-worker.js` script is performing. Since most of the action takes place during the service worker's install handler, and that is only executed the very first time you visit this page (unless the service worker script changes), it can be helpful to inspect the logging in a Chrome Incognito window.

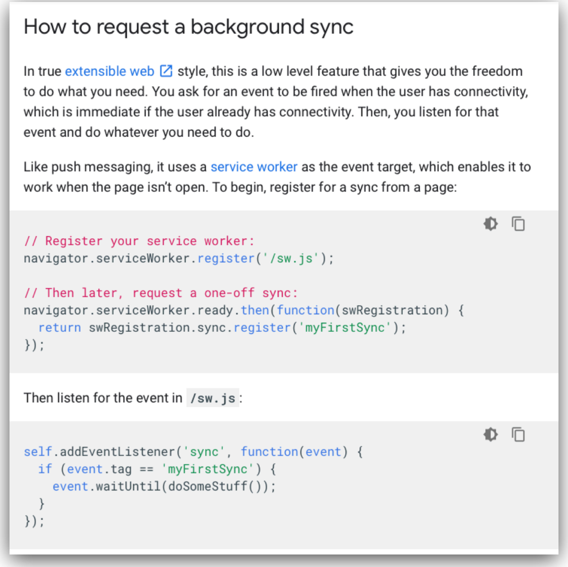
Some of the resources below have been pre-fetched, and some haven't. Try disabling your network connection and clicking on all the links. The resources that have been pre-fetched should be served from the Service Worker's cache and display as expected; the resource that hasn't been pre-fetched will be unavailable.

- pre_fetched.txt
- pre_fetched.html
- not_pre_fetched.txt

https://googlechrome.github.io/samples/service-worker/prefetch/

Background Sync

…defer until connectivity…



How to request a background sync

In true extensible web style, this is a low level feature that gives you the freedom to do what you need. You ask for an event to be fired when the user has connectivity, which is immediate if the user already has connectivity. Then, you listen for that event and do whatever you need to do.

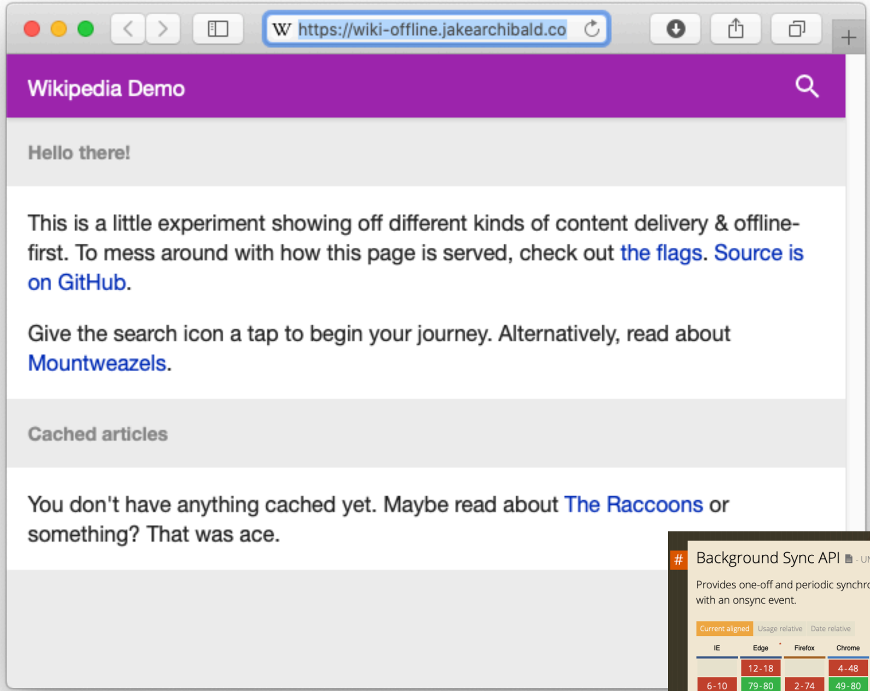
Like push messaging, it uses a service worker as the event target, which enables it to work when the page isn't open. To begin, register for a sync from a page:

```
// Register your service worker:
navigator.serviceWorker.register('/sw.js');

// Then later, request a one-off sync:
navigator.serviceWorker.ready.then(function(swRegistration) {
  return swRegistration.sync.register('myFirstSync');
});
```

Then listen for the event in `/sw.js`:

```
self.addEventListener('sync', function(event) {
  if (event.tag == 'myFirstSync') {
    event.waitUntil(doSomeStuff());
  }
});
```



https://wiki-offline.jakearchibald.co

Wikipedia Demo

Hello there!

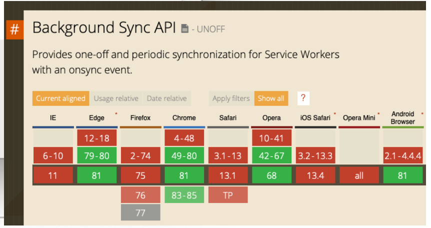
This is a little experiment showing off different kinds of content delivery & offline-first. To mess around with how this page is served, check out the flags. Source is on GitHub.

Give the search icon a tap to begin your journey. Alternatively, read about Mountweazels.

Cached articles

You don't have anything cached yet. Maybe read about The Raccoons or something? That was ace.

https://caniuse.com/#search=background



Background Sync API - UNOFF

Provides one-off and periodic synchronization for Service Workers with an onsync event.

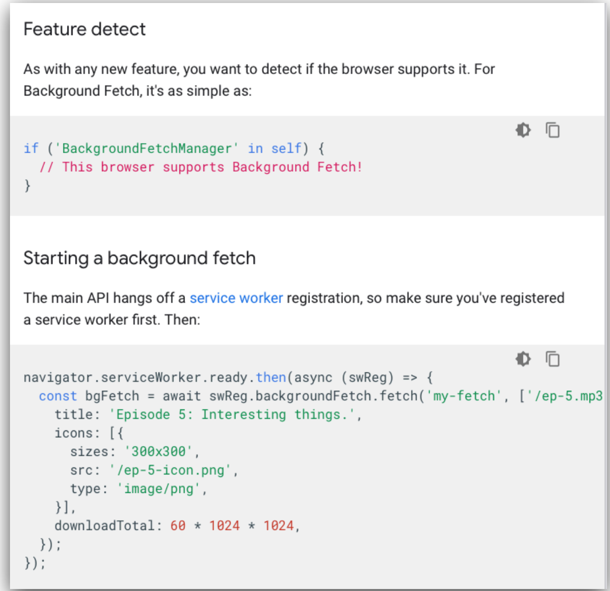
Current aligned
Usage relative
Date relative
Apply filters
Show all
?

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser
	12-18		4-48		10-41			
6-10	79-80	2-74	49-80	3.1-13	42-67	3.2-13.3		2.1-4.4.4
11	81	75	81	13.1	68	13.4	all	81
		76	83-85	TP				
		77						

https://developers.google.com/web/updates/2015/12/background-sync

https://wiki-offline.jakearchibald.com

Draft Only: Background Fetch



Feature detect

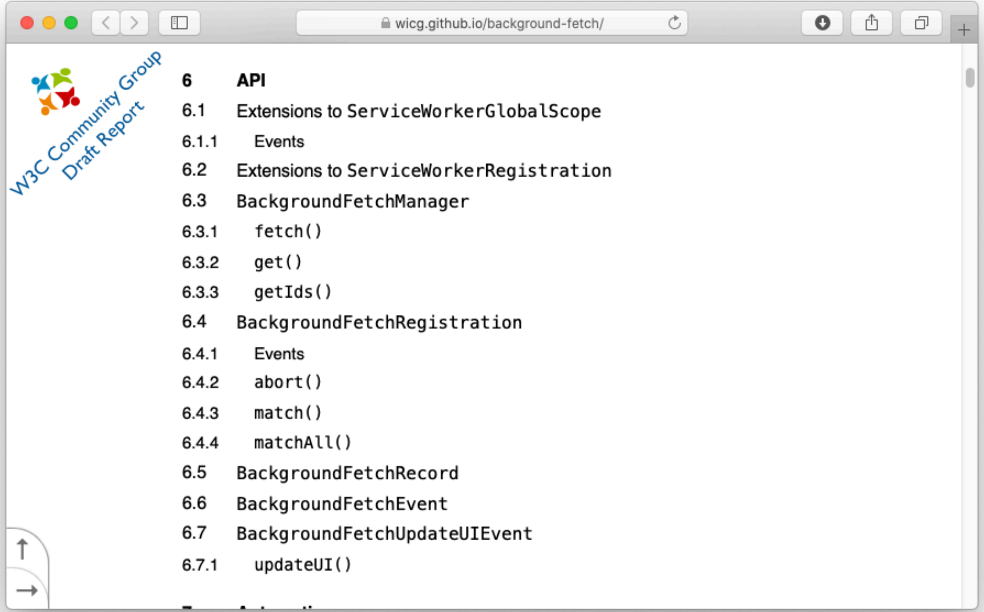
As with any new feature, you want to detect if the browser supports it. For Background Fetch, it's as simple as:

```
if ('BackgroundFetchManager' in self) {
  // This browser supports Background Fetch!
}
```

Starting a background fetch

The main API hangs off a service worker registration, so make sure you've registered a service worker first. Then:

```
navigator.serviceWorker.ready.then(async (swReg) => {
  const bgFetch = await swReg.backgroundFetch.fetch('my-fetch', ['/ep-5.mp3
    title: 'Episode 5: Interesting things.',
    icons: [{
      sizes: '300x300',
      src: '/ep-5-icon.png',
      type: 'image/png',
    }],
    downloadTotal: 60 * 1024 * 1024,
  });
});
```


wicg.github.io/background-fetch/
W3C Community Group Draft Report

https://wicg.github.io/background-fetch/

https://developers.google.com/web/updates/2018/12/background-fetch