

# Web-APIs

The missing manual

<https://giftkugel.github.io/talks/>  
or

<https://bit.ly/2JdoLoG>

## About me

**Simon Skoczylas**  
Senior Software Engineer  
**Karakun.**



**Karakun Developer Hub**

 *giftkugel*

 [www.skoczylas.net](http://www.skoczylas.net)



Karakun DevHub\_

@giftkugel

There was Adobe Flash (Thoughts on Flash)  
and Java Applets

Mark Zuckerberg: Our Biggest Mistake Was Betting Too Much On  
HTML5



# The Web-Browser as runtime environment with standardized APIs



Karakun DevHub\_

@giftkugel

# HTML5

<https://www.w3.org/TR/html5/>



# Web-APIs

<https://developer.mozilla.org/kab/docs/Web/API>



Karakun DevHub\_

@giftkugel

# Web-APIs I

*DOM*

*Fetch*

*Web Storage*

*Geolocation*

*Web Workers*

*Battery Status*

*PageVisibility*

*Navigator OnLine*

*Notifications*

*Vibration*

*Network Information*

*Web Share*

*Service Worker*

*Cache*

*Push*

*Background Sync*

*Web App Manifest*



# Web-APIs II

*Web Speech*

*Web Socket*

*Web Bluetooth*

*Web USB*

*Web Audio*

*Web Crypto*

*Payment Request*

*Credential Management*

*WebRTC*

*IndexedDB*

*Server-Sent Events*

*Server Timing*

*Navigation Timing*

*Generic Sensor*

*Streams*

*Gamepad*

*Canvas*

*Clipboard*

*Selection*

*Presentation*

*Fullscreen*

*Shape Detection*

*WebGL*

*WebVR*



# CSS Houdini

*Worklets*

*CSS Properties and Values*

*CSS Painting*

*CSS Layout*

*CSS Typed OM*

*CSS Animation Worklet*



## Current status

<https://platform-status.mozilla.org/>

<https://webkit.org/status/>

<https://www.chromestatus.com/features>

<https://developer.microsoft.com/en-us/microsoft-edge/platform/status/>

\*\*\*

<https://caniuse.com/>



# Feature detection

<https://html5test.com/>

<https://whatwebcando.today/>



# It's great to be a Web developer these days



Karakun DevHub\_

@giftkugel

# Battery Status API

[https://developer.mozilla.org/en-US/docs/Web/API/Battery\\_Status\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Battery_Status_API)

<https://www.w3.org/TR/battery-status/>



Karakun DevHub\_

@giftkugel

# Battery Status API

Provides information about the battery charge level

Notifies about changes

Current charge level: 100%

Charging: Yes

Removed from Mozilla and WebKit 😵



# Battery Status API

## Snippet

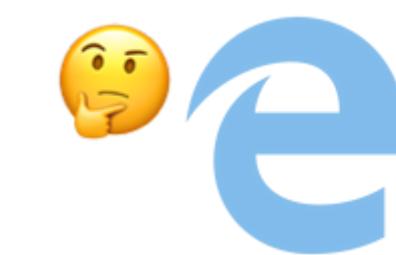
```
if (navigator.getBattery) {
    navigator.getBattery().then((battery) => {
        // battery.level (0 - 1)
        // battery.charging (true, false)

        battery.addEventListener('chargingchange', (event) => {});
        battery.addEventListener('levelchange', (event) => {});
        battery.addEventListener('chargingtimechange', (event) => {});
        battery.addEventListener('dischargingtimechange', f(event) => {});

    });
}
```



## Browser compatibility



# Page Visibility API

[https://developer.mozilla.org/en-US/docs/Web/API/Page\\_Visibility\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Page_Visibility_API)

<https://www.w3.org/TR/page-visibility-2/>



## Page Visibility API

Provides the visibility state of the current document

Notifies when the document becomes hidden or visible

Can be used to stop/pause long or expensive tasks

Current status: visible 😊



# Page Visibility API

## Snippet

```
// document.hidden (true, false)
// document.visibilityState (visible, hidden, prerender)
document.addEventListener('visibilitychange', (event) => {});
```



# Page Visibility API

## Demo

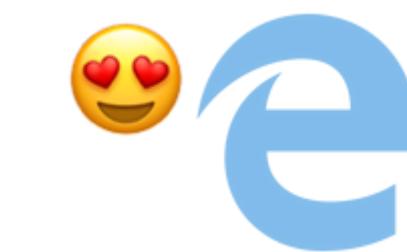
<http://daniemon.com/tech/webapps/page-visibility/>



Karakun DevHub\_

@giftkugel

## Browser compatibility



# NavigatorOnLine

aka Browser state

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

<https://www.w3.org/TR/html5/browsers.html#browser-state>



## NavigatorOnLine

Provides information about the connectivity status of the browser

Notifies when the connectivity changes

Current status: online 😊



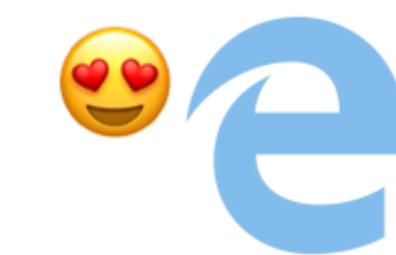
# NavigatorOnLine

## Snippet

```
// navigator.onLine (true, false)
window.addEventListener('offline', (event) => {});
window.addEventListener('online', (event) => {});
```



# Browser compatibility



# Notifications API

[https://developer.mozilla.org/en-US/docs/Web/API/Notifications\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Notifications_API)

<https://www.w3.org/TR/notifications/>



# Notifications API

Allows web pages to control system notifications

ServiceWorker got some additional abilities for notifications



# Notifications API

## Snippet

```
// Notification.permission (denied, granted, default)  
  
Notification.requestPermission().then(permission => {});  
  
new Notification("A fancy notification", options);
```



## Demo

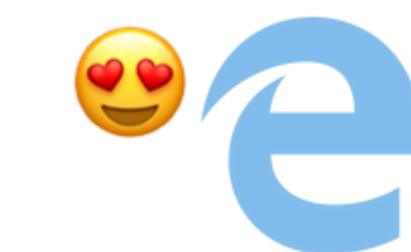


## Options

dir	data
lang	vibrate
body	renotify
badge	tag
icon	requireInteraction
image	actions*



## Browser compatibility



# Vibration API

[https://developer.mozilla.org/en-US/docs/Web/API/Vibration\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Vibration_API)

<https://www.w3.org/TR/vibration/>



Karakun DevHub\_

@giftkugel

## Vibration API

Offers the ability to access the vibration hardware

Does not work with display disabled ... 😕

Shakes your device!



## Snippet

```
navigator.vibrate(500);

navigator.vibrate([500, 250, 500, 250, 500, 250, 500, 250, 500, 250, 500]);

navigator.vibrate(0); // stop vibration
```



## Demo

[Vibrate after 5 secends](#) | [Stop](#)

<https://gearside.com/custom-vibration-patterns-mobile-devices/>



## Browser compatibility



Mostly supported by mobile browsers



# Network Information API

[https://developer.mozilla.org/en-US/docs/Web/API/Network\\_Information\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Network_Information_API)

<http://wicg.github.io/netinfo/>



Karakun DevHub\_

@giftkugel

# Network Information API

Provides information about the network connection

Effective network: 4g



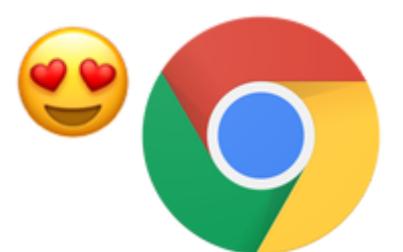
# Network Information API

## Snippet

```
// navigator.connection.type (bluetooth, cellular, ethernet, wifi, ...)  
// navigator.connection.downlink  
// navigator.connection.downlinkMax  
// navigator.connection.effectiveType (slow-2g, 2g, 3g, 4g)  
// navigator.connection.saveData (true, false)
```



## Browser compatibility



# Service Worker API

[https://developer.mozilla.org/en-US/docs/Web/API/Service\\_Worker\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API)

<https://www.w3.org/TR/service-workers/>



## Service Worker API

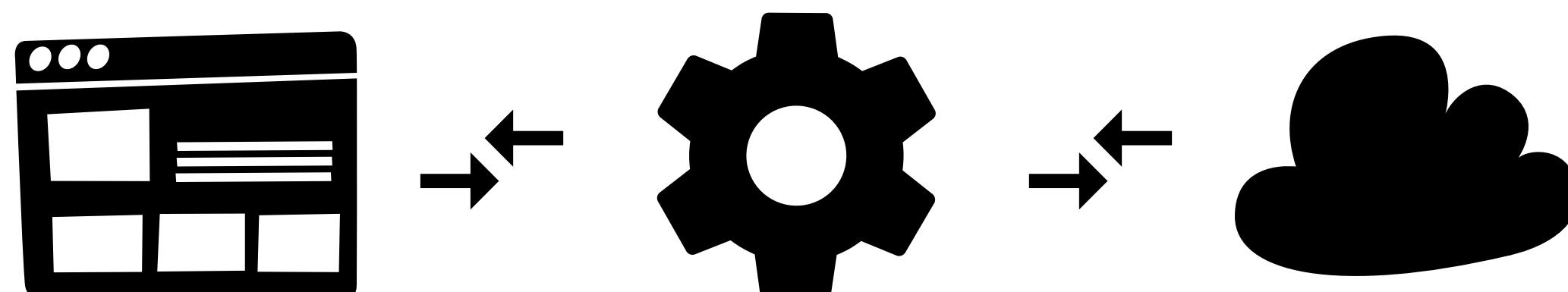
Service Worker is a script that runs in the background

*Provides: Cache API, Push API, Background Sync API*

As a Worker it can't access the DOM directly

Can communicate with the page via the [postMessage](#) interface

It's a programmable network proxy, allowing you to control how network requests are handled



# Enables Progressive Web Apps



# Service Worker API

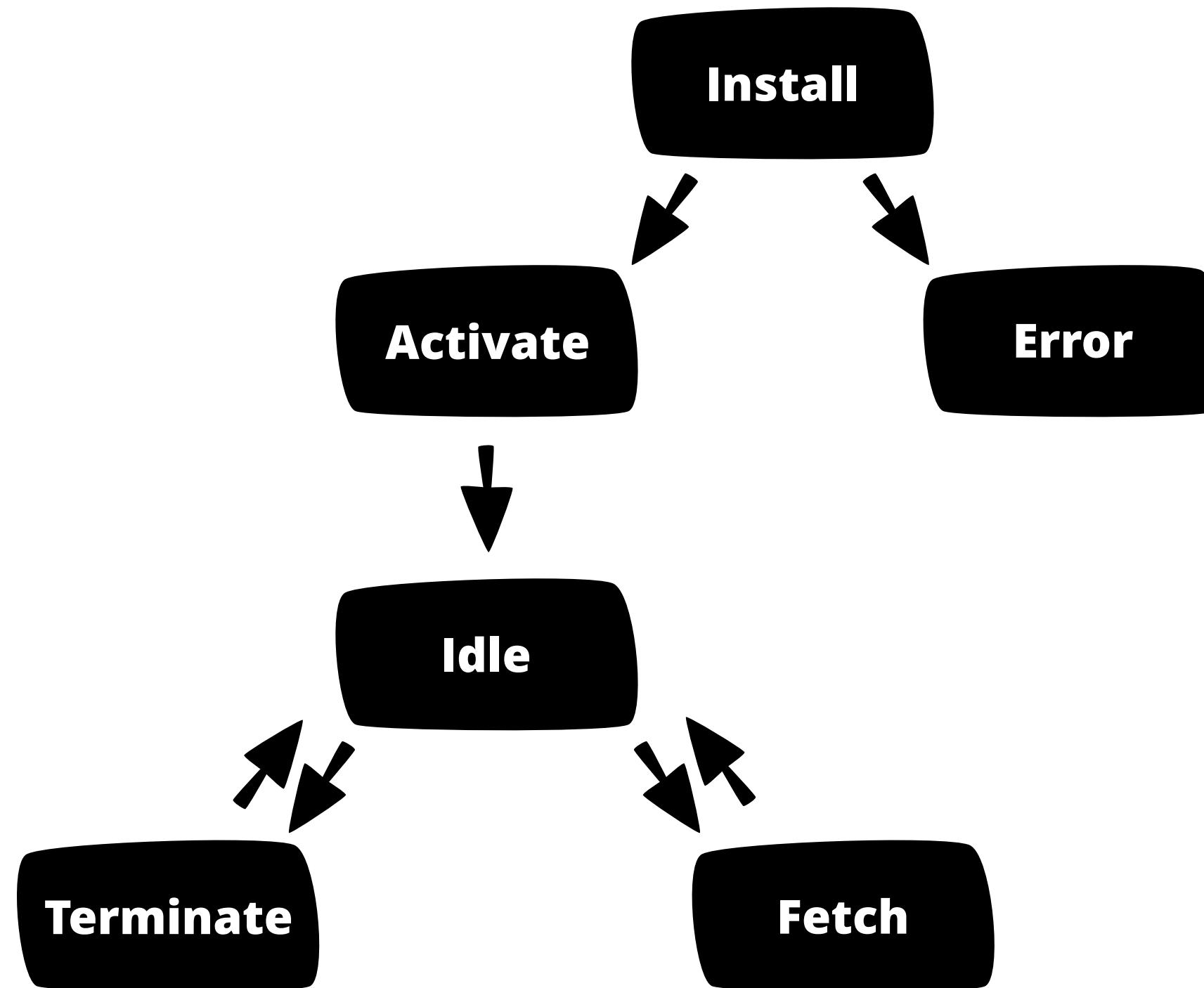
Requires HTTPS

Works without HTTPS on localhost (for development)

Requirement can be disabled through browser settings 🤪



# Lifecycle



## Register your service worker

```
if ('serviceWorker' in navigator) {  
    navigator.serviceWorker.register('/sw.js');  
}
```

Easy 😍



# Service Worker API

Maybe add a scope, wait for an event and handle result?

```
if ('serviceWorker' in navigator) {
  window.addEventListener('load', function() {
    navigator.serviceWorker.register('/sw.js', {scope: './'})
      .then((registration) => { ... })
      .catch((error) => { ... });
  });
}
```

ServiceWorkerRegistration



Karakun DevHub\_

@giftkugel

## this and self

Both points to the [ServiceWorkerGlobalScope](#)

Use *self*, because you never get a guarantee about *this* 😱



## Service Worker Events

```
self.addEventListener('install', (event) => { ... }); // InstallEvent  
  
self.addEventListener('activate', (event) => { ... }); // ExtendableEvent  
  
self.addEventListener('message', (event) => { ... }); // MessageEvent
```

aka ExtendableEvent



## Service Worker Functional Events

```
self.addEventListener('fetch', (event) => { ... }); // FetchEvent  
  
self.addEventListener('sync', (event) => { ... }); // SyncEvent  
  
self.addEventListener('push', (event) => { ... }); // PushEvent
```

aka ExtendableEvent



## Browser compatibility



<https://jakearchibald.github.io/isserviceworkerready/>



# Cache API

<https://developer.mozilla.org/de/docs/Web/API/Cache>

<https://www.w3.org/TR/service-workers/#cache-objects>



## Cache API

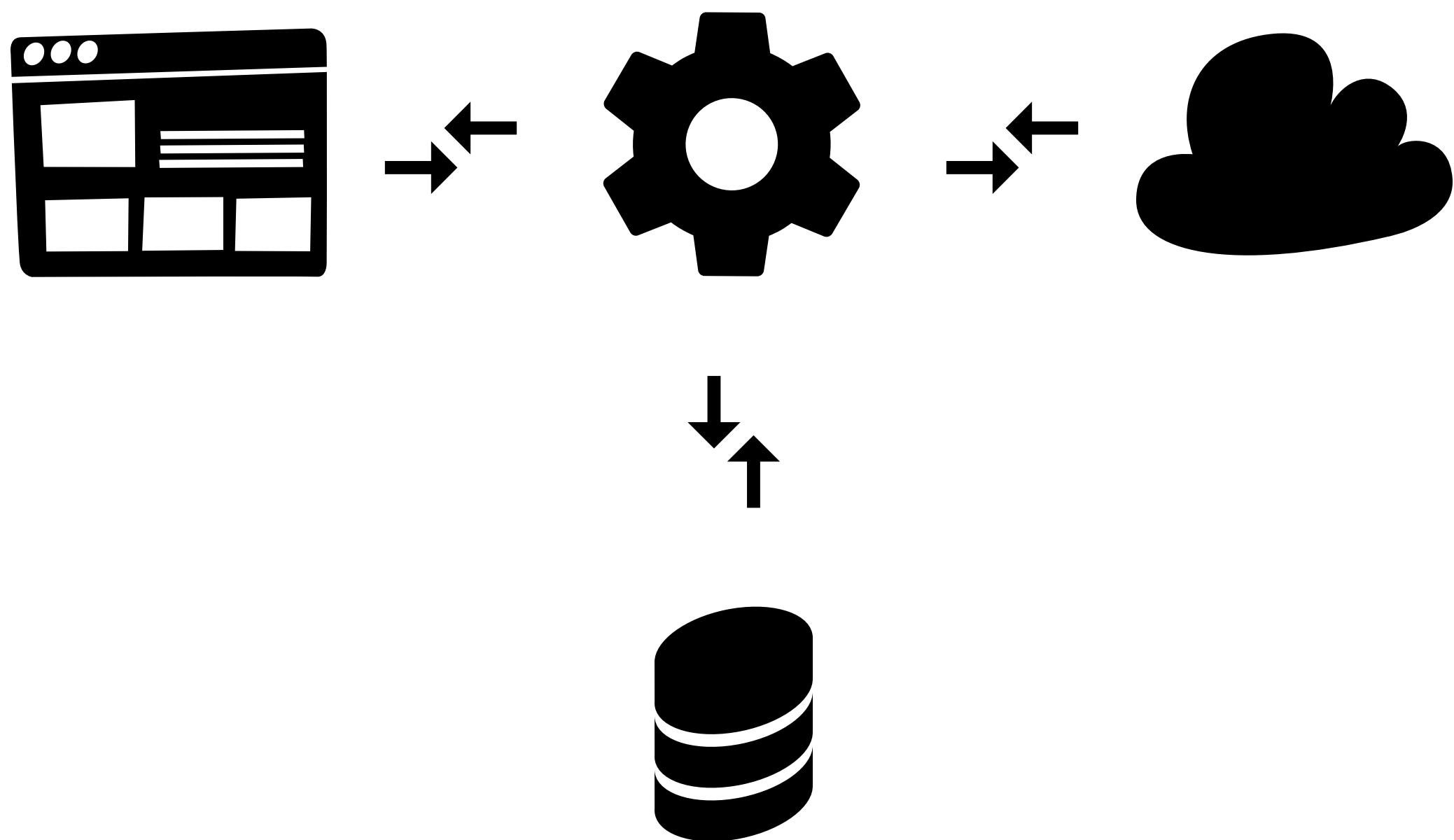
Provides a storage mechanism for Request/Response pairs

Actually for the Service Worker, but can be used outside

Consists of CacheStorage and Cache



# Cache API



## Snippet

```
caches.open(name).then((cache) => {}); // CacheStorage  
caches.has(name).then((boolean) => {}); // CacheStorage  
caches.keys(name).then((array) => {}); // CacheStorage  
caches.delete(name).then((boolean) => {}); // CacheStorage  
  
cache.match(request, options)  
cache.matchAll(request, options)  
  
cache.add(request)  
cache.addAll(requests)  
cache.put(request, response)  
  
cache.delete(request, options)  
cache.keys(request, options)
```



## Browser compatibility



Just as with the Service Worker API



# Push API

[https://developer.mozilla.org/de/docs/Web/API/Push\\_API](https://developer.mozilla.org/de/docs/Web/API/Push_API)

<https://www.w3.org/TR/push-api/>

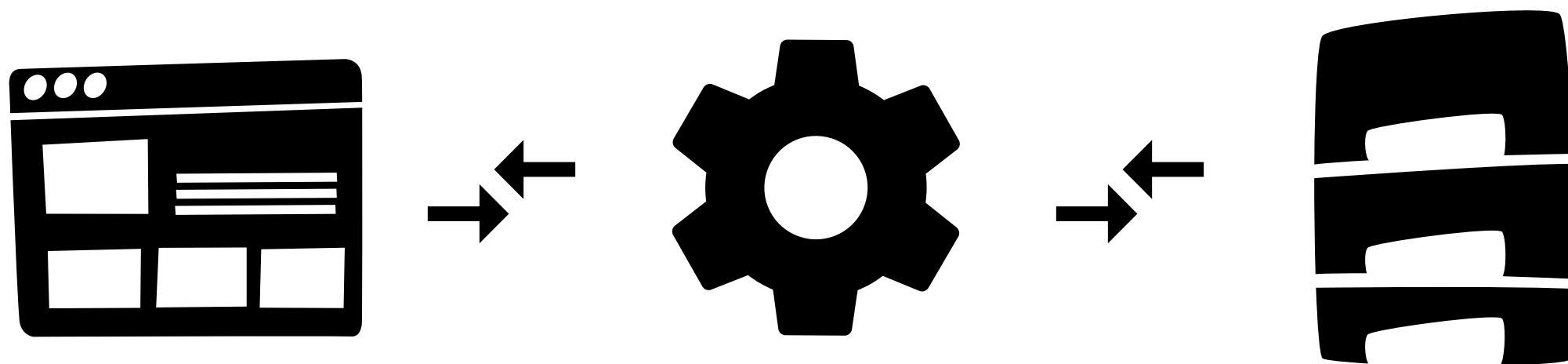


## Push API

Provides the ability to receive messages pushed from a server



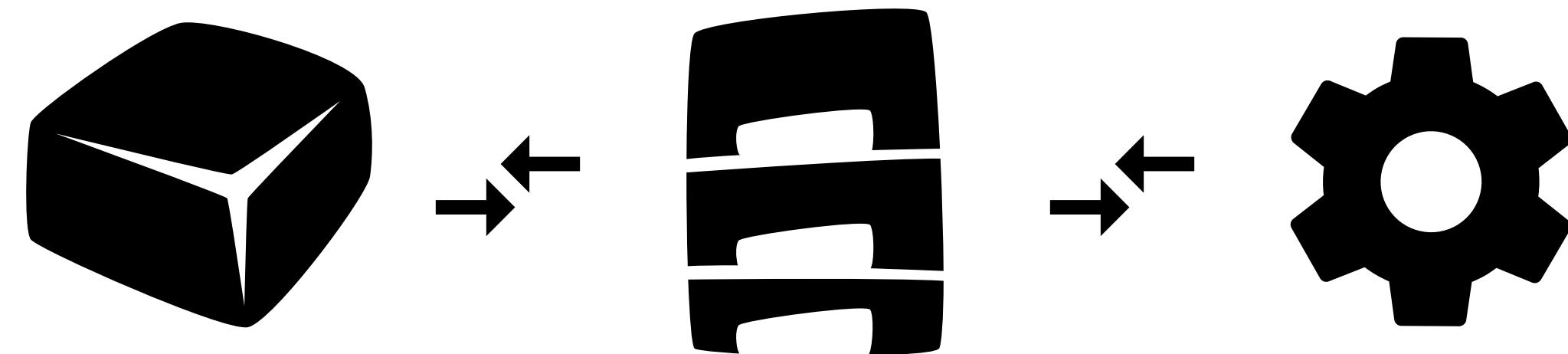
# Subscription



Page - Service Worker - Push Service



# Notification



Server - Push Service - Service Worker



## Snippet (Subscribe)

```
// const applicationServerKey = urlBase64ToUint8Array(applicationServerPublicKey);
navigator.serviceWorker.register('/sw.js').then((registration) => {
    // user interaction, then:
    registration.pushManager.getSubscription().then((subscription) => {
        if (subscription == null) {
            registration.pushManager.subscribe({
                userVisibleOnly: true,
                applicationServerKey: applicationServerKey
            })
            .then((newSubscription) => { /* store on server */ })
            .catch((error) => {});
        }
    }, (error) => {});
}) ;
```



## Snippet (Handle push in Service Worker)

```
self.addEventListener('push', (event) => {
  const text = event.data.text();
  const title = 'Push notification!';
  const options = {
    body: text
  };
  event.waitUntil(self.registration.showNotification(title, options));
});
```



# Web Push Protocol

<https://tools.ietf.org/html/draft-ietf-webpush-protocol-12>

## Server libraries

<https://github.com/web-push-libs>

Node.js, Java, PHP, Python, C#



## Snippet (Server-side with Node.js)

```
const webpush = require('web-push');

const vapidKeys = {
    publicKey: /* public key */,
    privateKey: /* private key */
}

webpush.setVapidDetails(
    'mailto:some@example.com',
    vapidKeys.publicKey,
    vapidKeys.privateKey
);

webpush.sendNotification(subscription, text);
```



## Demo

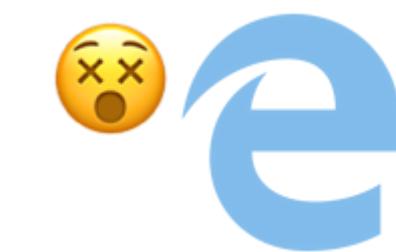
Request

Subscription

```
{"endpoint": "https://fcm.googleapis.com/fcm/send/cDU-wgC_wYc:APA91bFXV8VR_iYozNs5nG3o6tNHEOoukkqOKX_3CFxByDepjsE3U5MPigmGCvm7UPcc1ZQWgHSUZj_cjqCnCUgVwoq0ZehVHcssWwwqi2lq2hCVJ39MrS42QSPFqw0Vt3fqd1_dgk61", "expirationTime": null, "keys": {"p256dh": "BGE-6XYp6dd9mxXT-dXbWjlj2H-pzwdF3nll4PbvL7QQO70PK-7flN1bwjJ5BFqPunghTertWenZlZOWM7ucREw", "auth": "PggEHydU0q7zDUHDyS-wwQ"}}
```



## Browser compatibility



# Background Sync API

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

<http://wicg.github.io/BackgroundSync/spec/>



Karakun DevHub\_

@giftkugel

## Background Sync API

Provides a synchronization event for the Service Worker

Unlike [NavigatorOnline](#), the event can also be triggered when the page is not open

Will be triggered instant, or when the device regains connectivity



# Background Sync API

## Snippet (Registration)

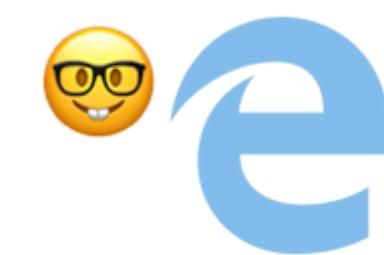
```
navigator.serviceWorker.register('/sw.js').then((registration) => {
  registration.sync.register('web-api-sync');
}) ;
```

## Snippet (Service Worker usage)

```
self.addEventListener('sync', (event) => {
  if (event.tag === 'web-api-sync') {
    /* do sync stuff */
  }
}) ;
```



## Browser compatibility



# Web Speech API

[https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Speech\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API)

<https://w3c.github.io/speech-api/>



Karakun DevHub\_

@giftkugel

# Web Speech API

Provides the ability to handle voice data

Text-to-Speech

Asynchronous Speech Recognition (network)



## Snippet (Text-to-speech)

```
const utterance = new SpeechSynthesisUtterance('Hallo Welt!');  
// lang, pitch, rate, voice, volume  
utterance.lang = 'de'  
speechSynthesis.speak(utterance);
```

## Snippet (Speech Recognition)

```
const grammar = '#JSGF V1.0; grammar ...';  
const recognition = new webkitSpeechRecognition();  
const speechRecognitionList = new webkitSpeechGrammarList();  
speechRecognitionList.addFromString(grammar, 1);  
recognition.grammars = speechRecognitionList;  
  
recognition.lang = 'en-US';  
recognition.interimResults = false;  
recognition.maxAlternatives = 1;  
  
recognition.onresult = (event) => {}  
recognition.start();
```



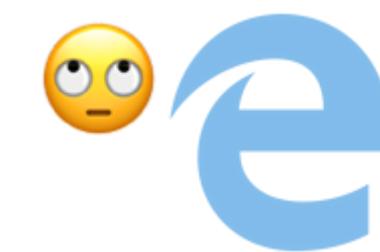
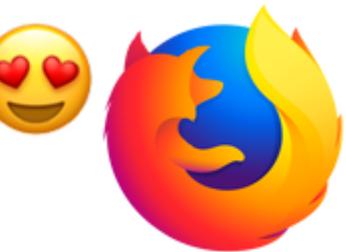
## Demo

<https://mdn.github.io/web-speech-api/speech-color-changer/>



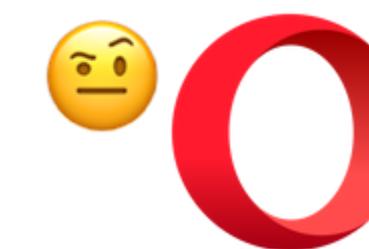
## Browser compatibility

Text-to-Speech



## Browser compatibility

Speech Recognition



# Web Bluetooth API

[https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Bluetooth\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Bluetooth_API)

<https://webbluetoothcg.github.io/web-bluetooth/>



Karakun DevHub\_

@giftkugel

# Web Bluetooth API

Provides the ability to connect and interact with Bluetooth Low Energy peripherals



# Web Bluetooth API

## Snippet

```
let options = {
  filters: [
    {services: ['heart_rate']}
  ],
  optionalServices: ['battery_service']
}

navigator.bluetooth.requestDevice([options]).then(function(bluetoothDevice) { ... })
```

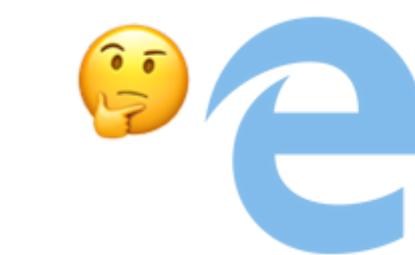
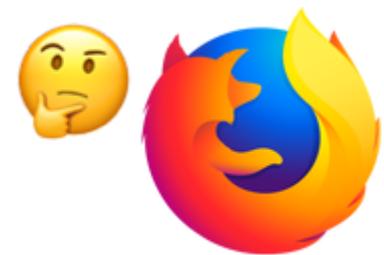
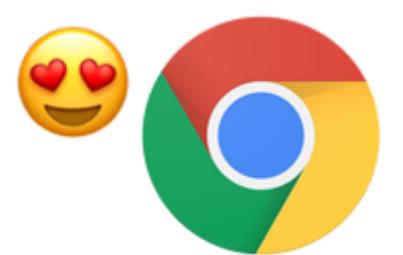


## Demo

<https://webbluetoothcg.github.io/demos/heart-rate-sensor/>



## Browser compatibility



# Web Share API

<https://developer.mozilla.org/en-US/docs/Web/API/Navigator/share>

<https://wicg.github.io/web-share/>



# Web Share API

Provides the possibility to use the native sharing mechanism

Share!

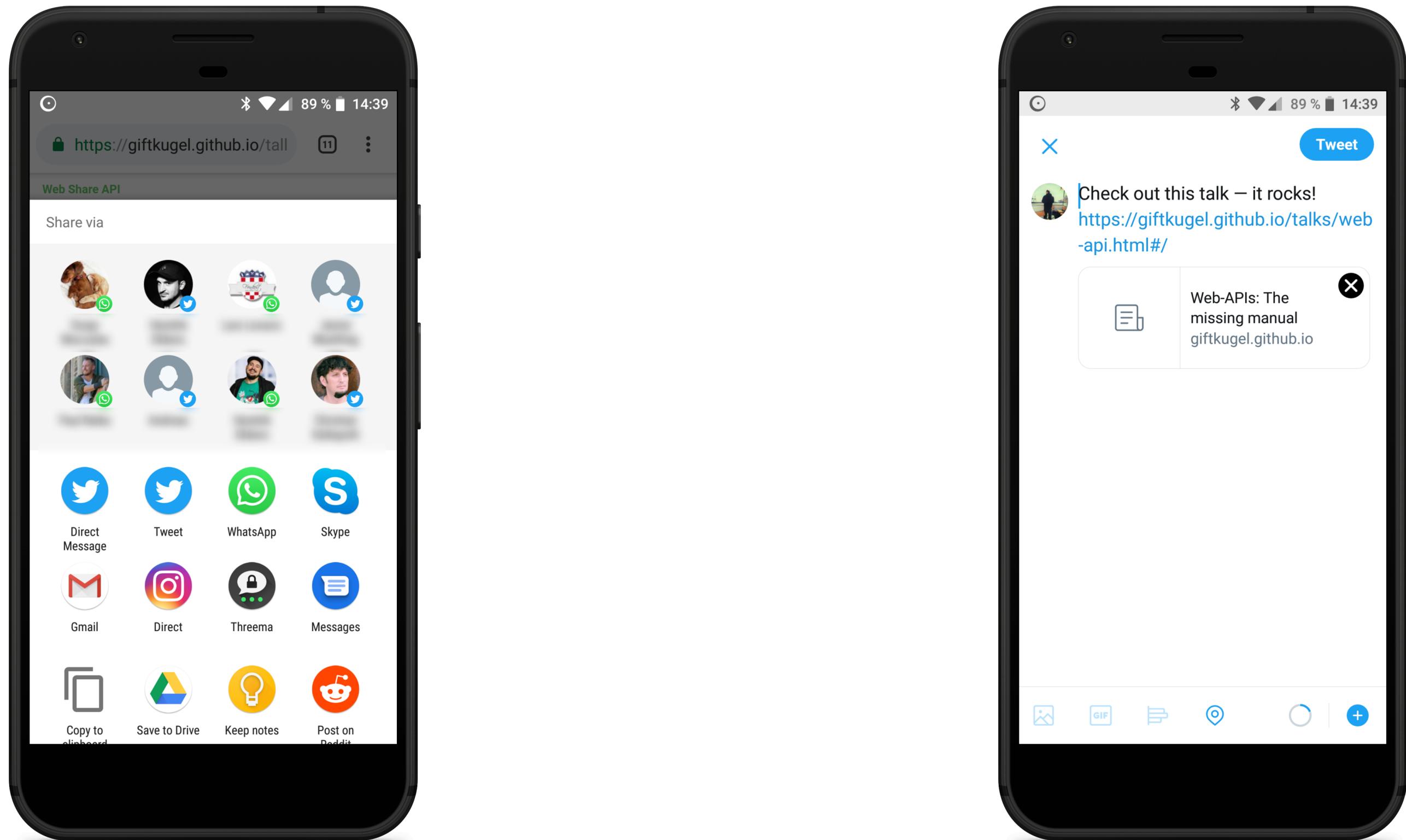


# Web Share API

## Snippet

```
navigator.share({
  title: 'Web-APIs',
  text: 'Check out this talk – it rocks!',
  url: 'https://giftkugel.github.io/talks/web-api.html#/'
})
.then(() => console.log('Yay!'))
.catch(error) => console.log('Doh', error));
```





Karakun DevHub\_

@giftkugel

## Browser compatibility

Currently only Chrome on Android



Thank you for listening

# Web development is cool!

Have fun with all the Web-APIs



Stay in touch @ [dev.karakun.com](https://dev.karakun.com)



Karakun DevHub\_

@giftkugel

## Legal stuff

### Images (CC0) taken from

<https://www.pexels.com/photo/camera-cctv-control-monitoring-274895/>  
<https://www.pexels.com/photo/silhouette-of-woman-raising-her-hands-615334/>  
<https://www.pexels.com/photo/man-in-white-shirt-using-macbook-pro-52608/>  
<https://www.pexels.com/photo/bind-blank-blank-page-business-315790/>  
<https://www.pexels.com/photo/building-construction-site-work-38293/>  
<https://www.pexels.com/photo/road-pavement-229014/>

### Phone mockups generated with

<https://mockuphone.com/#android>

### Icons

<https://material.io/icons/>  
<http://mariodelvalle.github.io/CaptainIconWeb/>  
<https://github.com/alrra/browser-logos>

