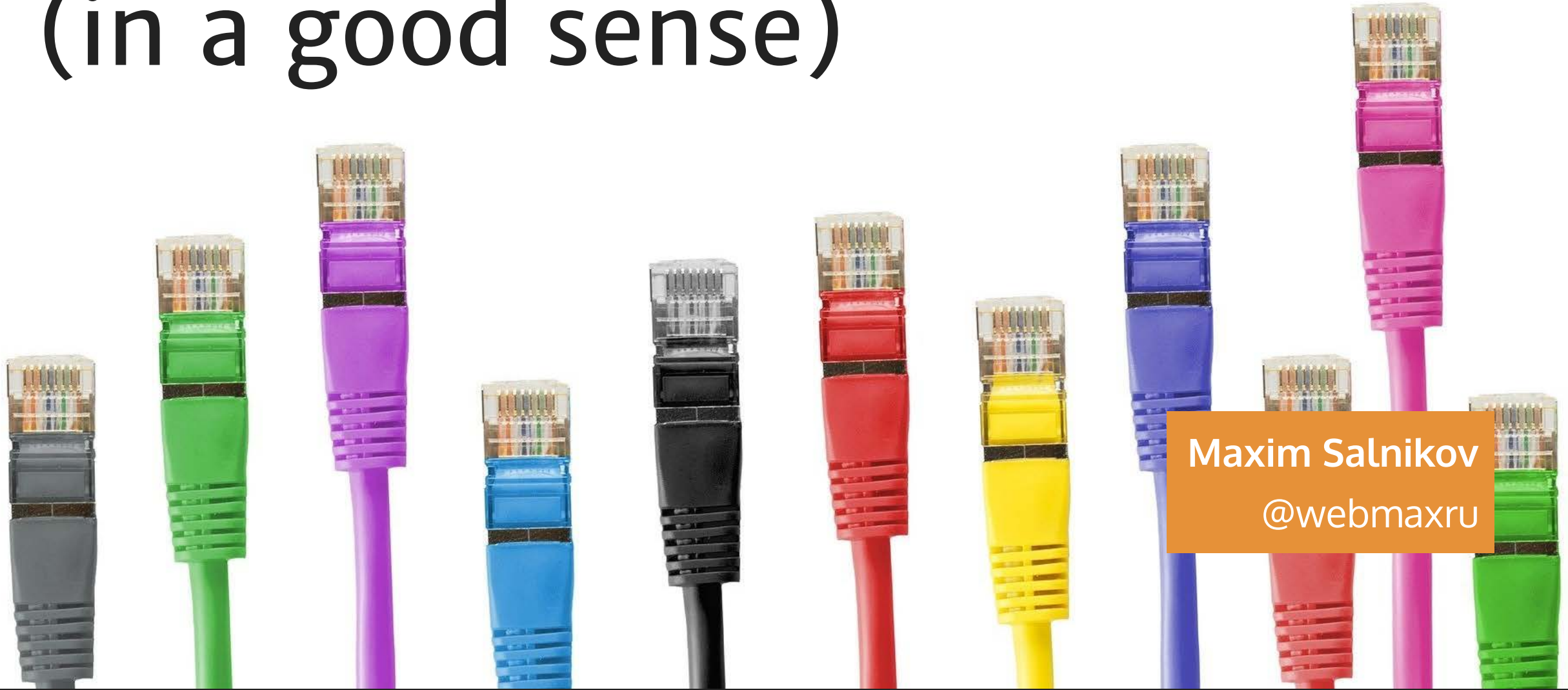


Taking your web app offline (in a good sense)



Maxim Salnikov
@webmaxru

“What is an offline-ready web application

And how to build it today

Maxim Salnikov

Developer Audience Lead at Microsoft



- **PWA Oslo / PWA London** meetups organizer
- **ngVikings / Mobile Era** conferences organizer
- **Google Dev Expert** in Web Tech / Capabilities & Installability

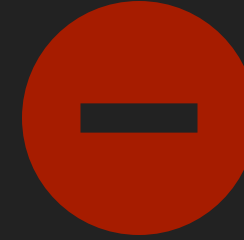


@webmaxru

Web as an app platform



- Evergreen browsers
- Versatile language
- Performant JS engines
- Excellent tooling
- Huge community



- Historically depends on the "connection status"



Proper offline-ready web app

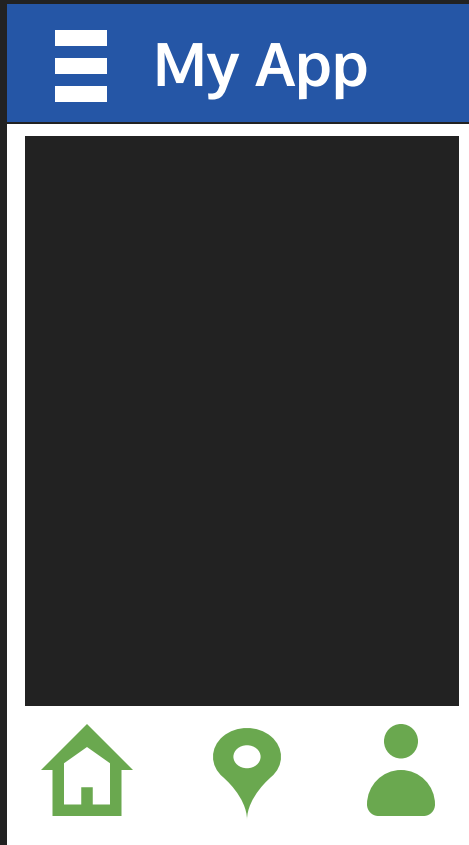
App itself	✓	Always available
Online runtime data	✓	Thoughtfully collected
Offline runtime data	✓	Safely preserved
Connection failures	✓	Do not break the flow
Updates	✓	Both explicit and implicit
Platform features	✓	For the win!

While keeping its web nature!



Application UI

Let's build an App shell



- Define assets
- Put in the cache
- Serve from the cache
- Manage versions



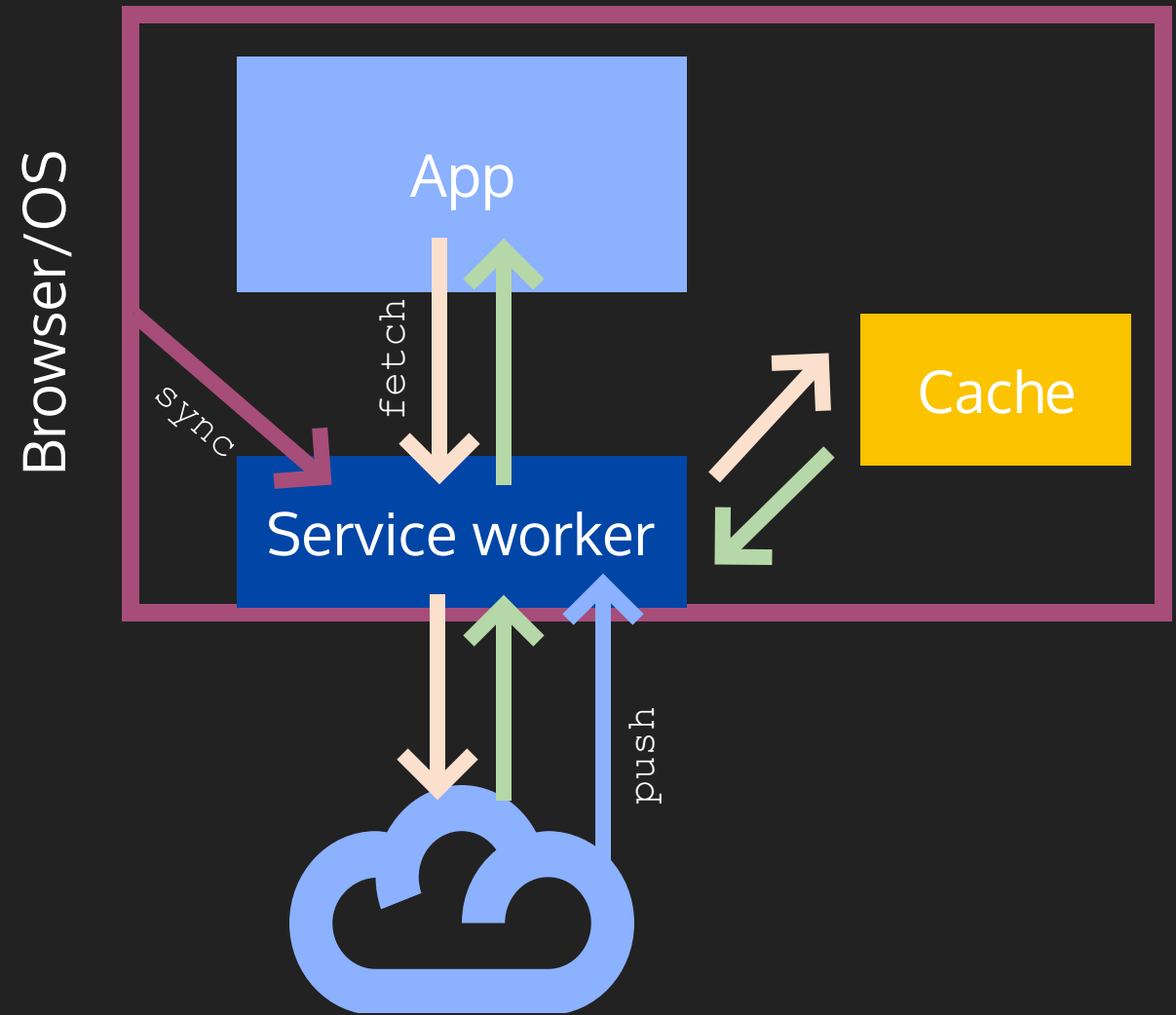
Service worker

Physically



Event-driven worker

Logically



Own service worker

handmade-service-worker.js

```
1 self.addEventListener('install', event => {
2     // Use Cache API to cache html/js/css
3 })
4
5 self.addEventListener('activate', event => {
6     // Clean the cache from the obsolete versions
7 })
8
9 self.addEventListener('fetch', event => {
10    // Serve assets from cache or network
11 })
```

It's only partially a joke



Sanket Sahu

@sanketsahu



Buys domain

Pushes wrong ServiceWorker to production

Buys another domain

7:07 AM · May 28, 2019 from [Bengaluru South, India](#) · [Twitter for iPhone](#)

225 Retweets **1.6K** Likes

<https://twitter.com/sanketsahu/status/1133223194888773632>

Because...

Redirects?

Cache

invalidation?

Spec updates?

Opaque
response?

Variable asset
names?

Versioning?

Fallbacks?

I see the old version!!!

Caching

Cache storage
space?

strategies?

Kill switch?

Routing?

Minimal required
cache update?

Feature
detection?

Fine-grained
settings?

Is there a helper?

- Define assets
- Put in the cache
- Serve from the cache
- Manage versions



Workbox

While having our own service worker



- Application shell
- Runtime caching
- Replaying failed network requests
- Offline Google Analytics
- Broadcasting updates

```
# Installing the Workbox Node module  
$ npm install workbox-build --save-dev
```

Configuration

workbox-build.js

```
1 // Sample configuration with the basic options
2 var workboxConfig = {
3   globDirectory: 'dist/',
4   globPatterns: [
5     '**/*.{txt,png,ico,html,js,json,css}'
6   ],
7   swSrc: 'src/workbox-service-worker.js',
8   swDest: 'dist/sw.js'
9 }
```

Source service worker

src/workbox-service-worker.js

```
1 import { precacheAndRoute } from "workbox-precaching";  
2  
3 // Precaching and setting up the routing  
4 precacheAndRoute(self.__WB_MANIFEST)
```

```
[  
  {  
    "url": "index.html",  
    "revision": "34c45cdf166d266929f6b532a8e3869e"  
  },  
  {  
    "url": "favicon.ico",  
    "revision": "b9aa7c338693424aae99599bec875b5f"  
  },  
  ...  
]
```



Caching, serving, managing versions

Build service worker

workbox-build.js

```
1 // We will use injectManifest mode
2 const {injectManifest} = require('workbox-build')
3
4 // Sample configuration with the basic options
5 var workboxConfig = {...}
6
7 // Calling the method and output the result
8 injectManifest(workboxConfig).then(({count, size}) =>
9     console.log(`Generated ${workboxConfig.swDest},
10     which will precache ${count} files, ${size} bytes
11     `))
```

Build flow integration

package.json

```
1 {  
2   "scripts": {  
3     "build-sw": "node workbox-build.js && npx rollup -  
4     "build":    "npm run build-app && npm run build-sw  
5   }  
6 }
```



Runtime application data

Intercepting requests

handmade-service-worker.js

```
1 self.addEventListener('fetch', event => {
2
3   if (event.request.url.indexOf('/api/breakingnews') !== -
4     event.respondWith(
5       // Network-First Strategy
6     )
7   } else if (event.request.url.indexOf('/api/archive') !==
8     event.respondWith(
9       // Cache-First Strategy
10    )
11  }
12 })
```

Routes and strategies

src/workbox-service-worker.js

```
1 import { registerRoute } from "workbox-routing";
2 import * as strategies from "workbox-strategies";
3
4 registerRoute(
5   new RegExp( '/api/breakingnews' ),
6   new strategies.CacheFirst()
7 );
```

```
1 registerRoute(
2   new RegExp( '/api/archive' ),
3   new strategies.CacheFirst({plugins: [...]}))
4 );
```

Strategies

- CacheFirst
- CacheOnly
- NetworkFirst
- NetworkOnly
- StaleWhileRevalidate

Plugins

- Expiration
- CacheableResponse
- BroadcastUpdate
- BackgroundSync
- ...your own plugin?



Save and sync offline actions

Background sync

- One-off event for offline -> online
- No active app tab required

main.js

```
1 navigator.serviceWorker.ready.then( swRegistration =>
2   return swRegistration.sync.register( 'postTweet' );
3 });
```

handmade-service-worker.js

```
1 self.addEventListener( 'sync', event => {
2   if (event.tag == 'postTweet') {
3     event.waitUntil(
4       // Do useful things...
5     );
6   }
7 });
```

src/workbox-service-worker.js

```
registerRoute(  
  /(http[s]?:\/\/)?(^[^\/\s]+\//)post-tweet/,  
  new strategies.NetworkOnly({plugins: [postTweetPlugin],  
    'POST'  
  })  
)
```

```
import {BackgroundSyncPlugin} from 'workbox-background-sync'
```

```
const postTweetPlugin =  
  new BackgroundSyncPlugin('tweetsQueue', {  
    maxRetentionTime: 24 * 60 // Max retry period  
  })
```



Schedule your updates

Periodic background sync

- App installed
- Online
- Original network
- Occasional interactions
- No cadence control
- Shortest break: 12 hours

Periodically run custom code without notifying user

main.js

```
1  const registration = await navigator.serviceWorker.re
2  if ('periodicSync' in registration) {
3    try {
4      registration.periodicSync.register('refreshTweets
5      // An interval of one day.
6      minInterval: 24 * 60 * 60 * 1000,
7      });
8    } catch (error) {
9      // PBS cannot be used.
10   }
11 }
```

src/service-worker.js







```
1 self.addEventListener('periodicsync', (event) => {
2   if (event.tag === 'refreshTweets') {
3     event.waitUntil(
4       fetchNewTweets() // Maybe
5     );
6   }
7 });
```

- Network connection type?
- Data saver mode?
- Available storage space?

Debugging background services

Chromium -> DevTools -> Application

Background Services

-  Background Fetch
-  Background Sync
-  Notifications
-  Payment Handler
-  Periodic Background Sync
-  Push Messaging

Show events from other domains

#	Timestamp	Event	Origin	SW...	Instance ID
1	2020-05-28 14:01:52.224	Dispatched periodicsync event	https://www.webmax.ru/	/	test-tag-from-devtools
2	2020-05-28 14:01:52.442	Dispatched periodicsync event	https://www.webmax.ru/	/	test-tag-from-devtools
3	2020-05-28 14:01:53.192	Dispatched periodicsync event	https://www.webmax.ru/	/	test-tag-from-devtools
4	2020-05-28 14:01:53.404	Dispatched periodicsync event	https://www.webmax.ru/	/	test-tag-from-devtools
5	2020-05-28 14:01:53.607	Dispatched periodicsync event	https://www.webmax.ru/	/	test-tag-from-devtools

A collection of various wrenches and tools is displayed on a blue, textured background. The tools are arranged in several rows. The top row features a variety of open-end wrenches in different sizes. Below this, there are more wrenches, some with circular ends. In the middle-right section, a set of six small open-end wrenches is mounted on a metal bar. To the right of this bar, a pair of calipers is visible. At the bottom right, there is a red and black tool tray containing several small bits or screws. The overall scene is a well-organized display of mechanical tools.

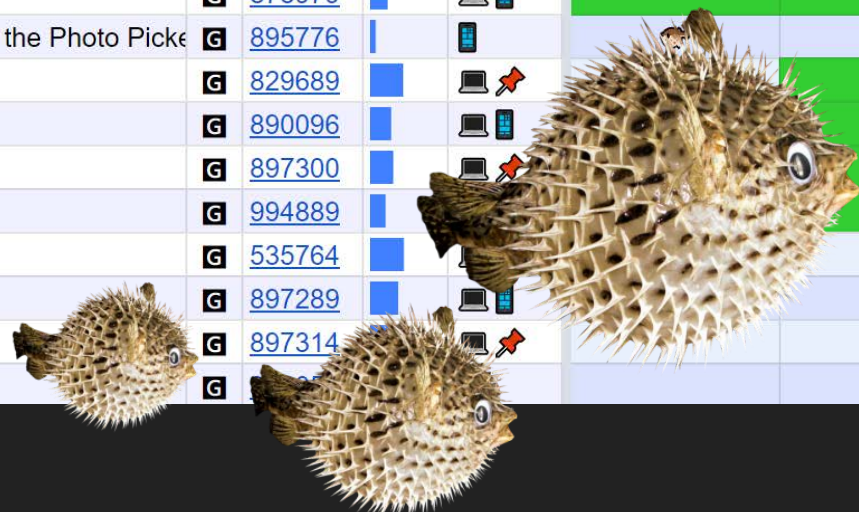
More platform tools

- Background Sync API
- Background Fetch API
- Native File System API
- Badging API
- Contact Picker API
- Notification Triggers API

+ more than 100 other APIs

Last updated: 5/27/2020, 11:11:08 AM

				84 (beta)	85 (canary)	86	87	88
Origin Trial (details)								
WakeLock API - Screen Lock	I	257511						
Web NFC.	I	520391						
Web OTP API	G	670299						
Notification Triggers (a.k.a. Scheduled Notification AP	G	891339						
Implement the Content Index API	G	973844						
Backpressure for WebSockets (WebSocket Streams)	G	840253						
Native File System API	G	853326						
Web Serial API	G	884928						
Pointer Lock Unadjusted Movement option	G	982379						
Dev Trial - behind a flag (details)								
Ambient Light Sensor API based on Generic Sensor A	I	606766						
User Idle Detection API	G	878979						
Create a Photo/Video picker similar to the Photo Picker	G	895776						
File Handling	G	829689						
WebHID API	G	890096						
Window Placement API	G	897300						
Screen Enumeration API	G	994889						
Font Enumeration API	G	535764						
Raw Clipboard Access API	G	897289						
Tabbed application mode for PWAs	G	897314						
Font Table Access API	G							



Web platform today

- Full-fledged application platform
- Offline-ready mechanisms are in production
- Awesome tools are available
- User experience & security is the key

And this is just the beginning!

aka.ms/pwa-slack

- 2000+ developers
- Major browsers/frameworks/libs reps

Thank you!

Maxim Salnikov

@webmaxru

Questions?

Maxim Salnikov

@webmaxru