

# **Aplicaciones Web sin Conexión**

## **Service Workers**

Departamento LSI

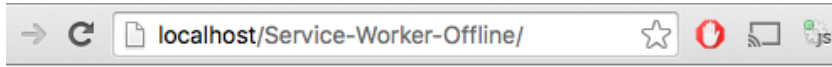
Juanan Pereira <[juanan.pereira@ehu.es](mailto:juanan.pereira@ehu.es)>

# Aplicaciones Web sin Conexión

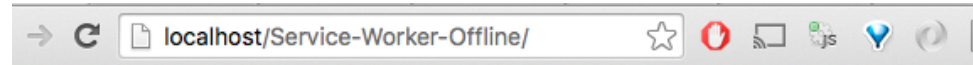


¿Podríamos llegar a tener acceso a otro contenido, por ejemplo, un test, **sin tener conexión**, permitirle al usuario hacer el test y sincronizarlo cuando volvamos a estar online?

# Aplicaciones Web sin Conexión



Nice! You are currently online.

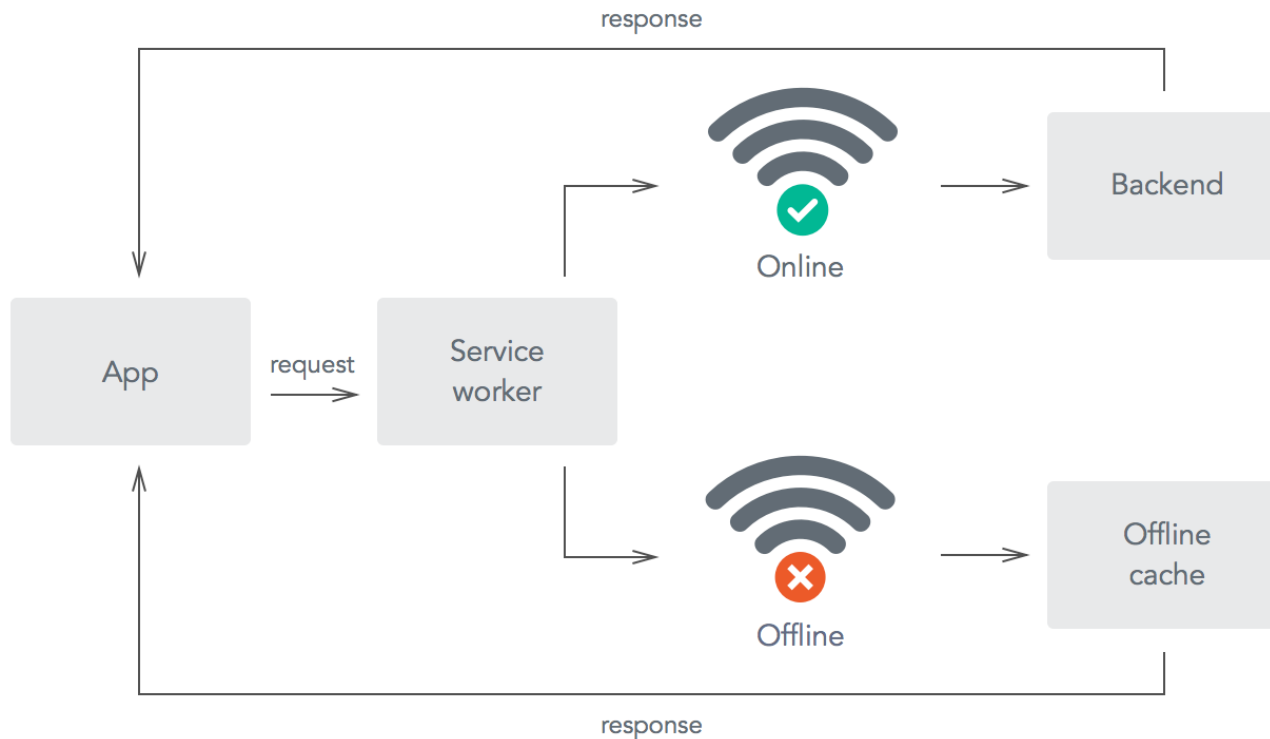


Boo! You don't have an internet connection.

Please check your network connection and try again.

# Aplicaciones Web sin Conexión

**Service Workers** > AppCache



# Aplicaciones Web sin Conexión

**1 Instalar Service Worker + añadir página offline a caché**

**2 Si el usuario intenta navegar sin conexión, devolver página cacheada**

**3 Si el usuario intenta navegar con conexión, continuar como siempre**

# Aplicaciones Web sin Conexión

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Home Page</title>
6 </head>
7 <body>
8   <div style="text-align:center; margin-top:40px;">
9     
10    <p>Nice! You are currently online.</p>
11  </div>
12 </body>
13 </html>
```

## Instalar Service Worker + añadir página offline a caché

### service-worker-register.js

```
1 <script>
2 // Register the service worker
3 if ('serviceWorker' in navigator) {
4     navigator.serviceWorker.register('/service-worker.js').then(function(registration) {
5         // Registration was successful
6         console.log('ServiceWorker registration successful with scope: ', registration.scope);
7     }).catch(function(err) {
8         // registration failed :(
9         console.log('ServiceWorker registration failed: ', err);
10    });
11 }
12 </script>
```

## Instalar Service Worker + añadir página offline a caché

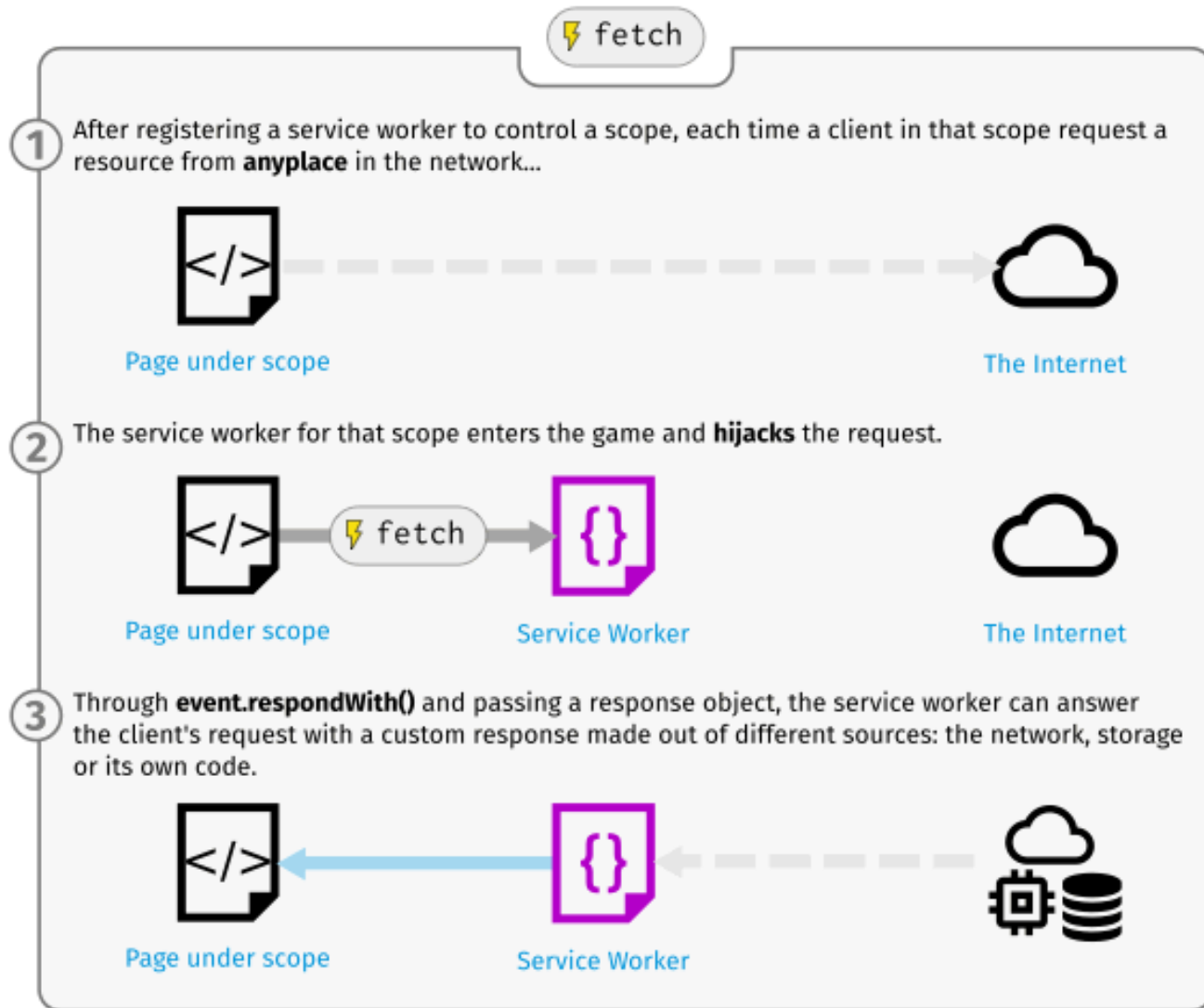
```
1  'use strict';
2
3  var cacheVersion = 1;
4  var currentCache = {
5    offline: 'offline-cache' + cacheVersion
6  };
7  const offlineUrl = 'offline-page.html';
8
9  this.addEventListener('install', event => {
10    event.waitUntil(
11      caches.open(currentCache.offline).then(function(cache) {
12        return cache.addAll([
13          './img/offline.svg',
14          offlineUrl
15        ]));
16      })
17    );
18  });
```

service-worker-add-offline-page.js hosted with ❤️ by GitHub

[view raw](#)



## Detectar navegación ONLINE | OFFLINE. El evento FETCH.



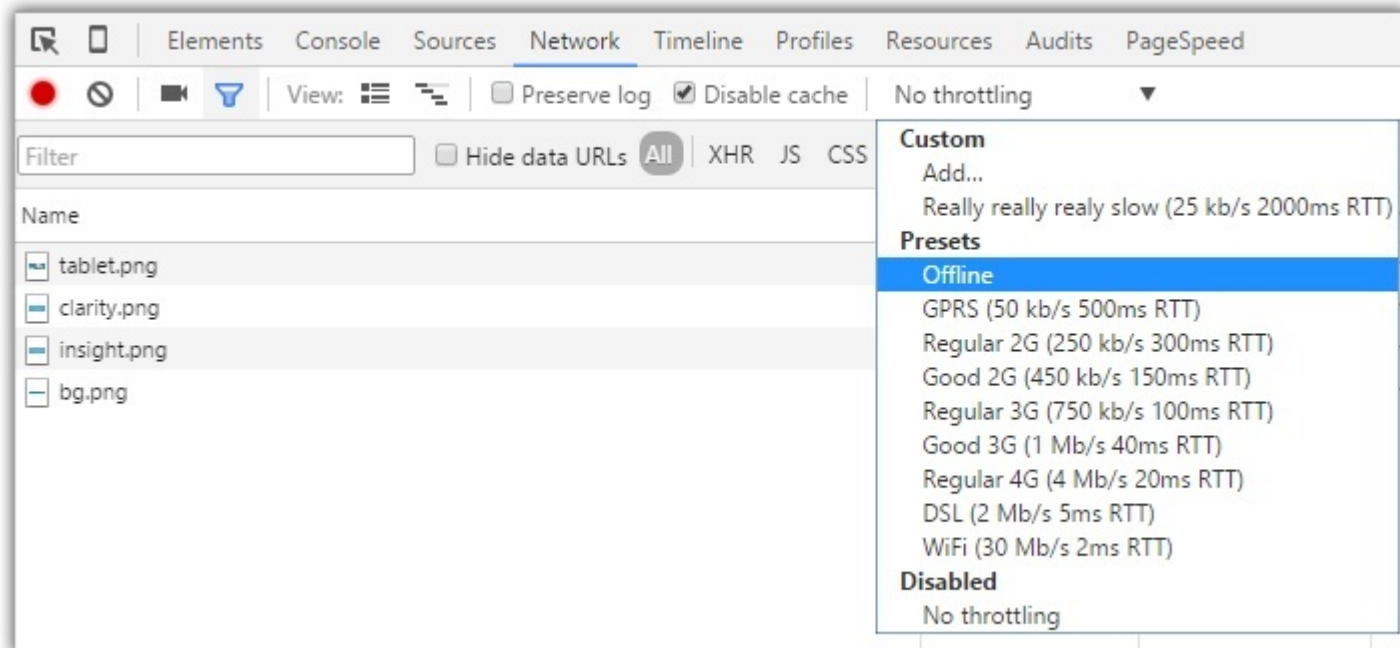
## Si el usuario intenta navegar sin conexión, devolver página cacheada

```
1  this.addEventListener('fetch', event => {
2    // request.mode = navigate isn't supported in all browsers
3    // so include a check for Accept: text/html header.
4    if (event.request.mode === 'navigate' || (event.request.method === 'GET' && event.request.headers.get('accept').includes('text/html'))) {
5      event.respondWith(
6        fetch(event.request.url).catch(error => {
7          // Return the offline page
8          return caches.match(offlineUrl);
9        })
10     );
11   }
12   else{
13     // Respond with everything else if we can
14     event.respondWith(caches.match(event.request)
15       .then(function (response) {
16         return response || fetch(event.request);
17       })
18     );
19   }
20 });
```

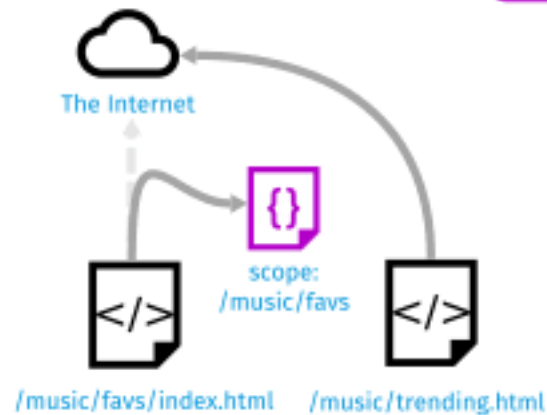
service-worker-offline-page.js hosted with ❤️ by GitHub

[view raw](#)

## Simulando un dispositivo OFFLINE



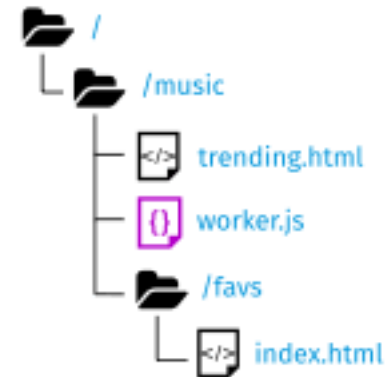
## Important!



The service worker will catch requests from the clients **under scope** only.



The service worker must be served over **https**.



The **max scope** for a service worker is the location of the worker.



# DEBUGGING

← → ↻

chrome://serviceworker-internals

Scope: <https://www.duolingo.com/>

Registration ID: 6

Active worker:

Installation Status: ACTIVATED

Running Status: STOPPED

Script: <https://www.duolingo.com/gcm-service-worker.js>

Version ID: 731

Renderer process ID: -1

Renderer thread ID: -1

DevTools agent route ID: -2

Log:

Unregister

Start

Scope: [https://www.google.com/\\_/chrome/](https://www.google.com/_/chrome/)

Registration ID: 1

Active worker:

Installation Status: ACTIVATED

Running Status: RUNNING

Script: [https://www.google.com/\\_/chrome/newtab-serviceworker.js](https://www.google.com/_/chrome/newtab-serviceworker.js)

Version ID: 1904

Renderer process ID: 1175

Renderer thread ID: 22559

DevTools agent route ID: 3

Log:

Stop

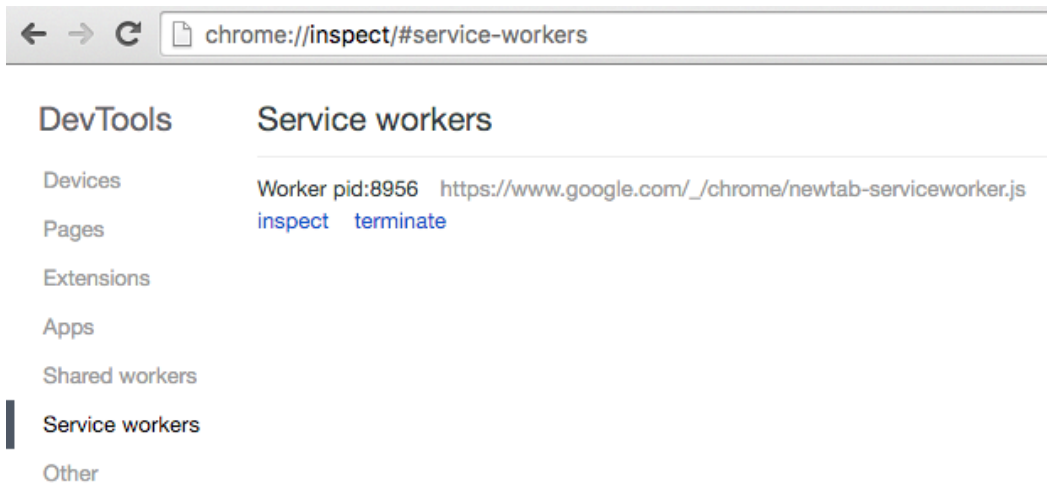
Push

Inspect

Unregister

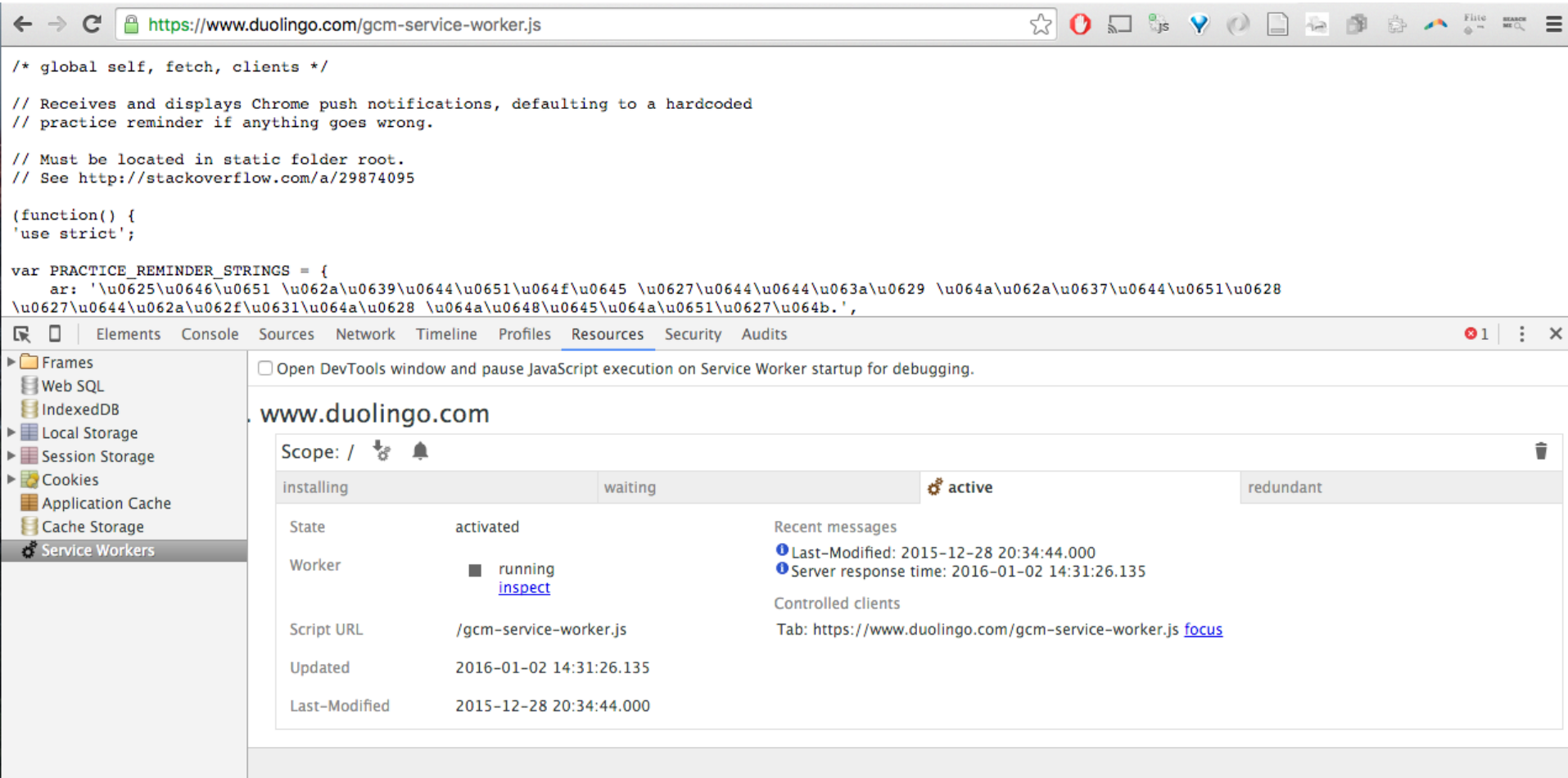


# DEBUGGING





# DEBUGGING



## Referencias Utilizadas

<http://deanhume.com/home/blogpost/create-a-really--really-simple-offline-page-using-service-workers/10135>  
<https://www.chromium.org/blink/serviceworker/service-worker-faq>  
<http://blog.lamplightdev.com/2015/01/06/A-Simple-ServiceWorker-App/>  
[https://developer.mozilla.org/en-US/docs/Web/API/Service\\_Worker\\_API/Using\\_Service\\_Workers](https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API/Using_Service_Workers)  
<https://auth0.com/blog/2015/10/30/creating-offline-first-web-apps-with-service-workers/>  
[https://infrequently.org/14/html5devconf\\_oct/#39](https://infrequently.org/14/html5devconf_oct/#39)

## Cosas que se han quedado en el tintero

<https://developers.google.com/web/fundamentals/getting-started/push-notifications/?hl=en>