**Service Worker API**

Los Service workers actúan esencialmente como proxy servers asentados entre las aplicaciones web, el navegador y la red (cuando está accesible). Están destinados, entre otras cosas, a permitir la creación de experiencias offline efectivas, interceptando peticiones de red y realizando la acción apropiada si la conexión de red está disponible y hay disponibles contenidos actualizados en el servidor. También permitirán el acceso a notificaciones tipo push y APIs de sincronización en segundo plano.

[**Descarga, instalación y activación**](https://developer.mozilla.org/es/docs/Web/API/Service_Worker_API#Descarga_instalaci%C3%B3n_y_activaci%C3%B3n)

En este punto, su service worker observará el siguiente ciclo de vida:

1. Descarga
2. Instalación
3. Activación

El service worker se descaga inmediatamente cuando un usuario accede por primera vez a un sitio controlado por el mismo.

Después de esto se descarga cada 24 horas aproximadamente. Se puede descargar con más frecuencia, pero **debe** ser descargado cada 24 horas para prevenir que una mala programación sea molesta durante mucho tiempo.

**ServiceWorkerContainer.register()**

[Parámetros](https://developer.mozilla.org/es/docs/Web/API/ServiceWorkerContainer/register#Par%C3%A1metros)

ServiceWorkerContainer.register(scriptURL, options)

.then(**function**(ServiceWorkerRegistration) { ... });

scriptURL

La URL del script de trabajador de servicio.

options Optional

Un objeto que contiene opciones de registro. Las opciones disponibles actualmente son:

* alcance: [USVString](https://developer.mozilla.org/es/docs/Web/API/USVString" \o "La documentación acerca de este tema no ha sido escrita todavía . ¡Por favor  considera contribuir !" \t "_blank) representa una URL que define el alcance de registro de un “service worker”; es decir, qué rango de URL puede controlar un “service worker”. Esto es generalmente una URL relativa. El valor predeterminado es la URL que obtendría si resolviera ‘./’ utilizando la ubicación de la página web como base. No es, como se cree comúnmente, relativo a la ubicación del “service worker”. Vea la sección de Ejemplos para más información sobre cómo funciona.

**Ejemplo**

**if** ('serviceWorker' **in** navigator) {

// Register a service worker hosted at the root of the

// site using the default scope.

navigator.serviceWorker.register('/sw.js').then(**function**(registration) {

console.log('Service worker registration succeeded:', registration);

}).catch(**function**(error) {

console.log('Service worker registration failed:', error);

});

} **else** {

console.log('Service workers are not supported.');

}

**CacheStorage.open()**

The **open()** method of the [CacheStorage](https://developer.mozilla.org/en-US/docs/Web/API/CacheStorage" \o "The CacheStorage interface represents the storage for Cache objects." \t "_blank) interface returns a [Promise](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise" \o "The Promise object represents the eventual completion (or failure) of an asynchronous operation, and its resulting value." \t "_blank) that resolves to the [Cache](https://developer.mozilla.org/en-US/docs/Web/API/Cache) object matching the cacheName.

**Ejemplo**

**var** cachedResponse = caches.match(event.request).catch(**function**() {

**return** fetch(event.request);

}).then(**function**(response) {

caches.open('v1').then(**function**(cache) {

cache.put(event.request, response);

});

**return** response.clone();

}).catch(**function**() {

**return** caches.match('/sw-test/gallery/myLittleVader.jpg');

});

**Cache**

La \*\*Cache\*\*interfaz proporciona un mecanismo de almacenamiento para [Request](http://fetch.spec.whatwg.org/#request)/ [Response](http://fetch.spec.whatwg.org/#response)pares de objetos que se almacenan en caché, por ejemplo, como parte del [ServiceWorker](https://developer.mozilla.org/en-US/docs/Web/API/ServiceWorker" \o "La interfaz ServiceWorker de la API ServiceWorker proporciona una referencia a un trabajador de servicio.  Se pueden asociar múltiples contextos de navegación (por ejemplo, páginas, trabajadores, etc.) con el mismo trabajador de servicio, cada uno a través de un objeto único de ServiceWorker." \t "_blank)ciclo de vida. Tenga en cuenta que la Cacheinterfaz está expuesta a ámbitos con ventanas, así como a los trabajadores. No tiene que usarlo junto con los trabajadores del servicio, aunque esté definido en la especificación del trabajador del servicio.

**Métodos**[**Sección**](https://developer.mozilla.org/en-US/docs/Web/API/Cache#Methods)

[Cache.match(request, options)](https://developer.mozilla.org/en-US/docs/Web/API/Cache/match)

Devuelve un [Promise](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise" \o "El objeto Promise representa la finalización (o falla) eventual de una operación asincrónica y su valor resultante." \t "_blank)que resuelve la respuesta asociada con la primera solicitud coincidente en el Cacheobjeto.

[Cache.matchAll(request, options)](https://developer.mozilla.org/en-US/docs/Web/API/Cache/matchAll)

Devuelve un [Promise](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise" \o "El objeto Promise representa la finalización (o falla) eventual de una operación asincrónica y su valor resultante." \t "_blank)que resuelve una matriz de todas las solicitudes coincidentes en el Cacheobjeto.

[Cache.add(request)](https://developer.mozilla.org/en-US/docs/Web/API/Cache/add)

Toma una URL, la recupera y agrega el objeto de respuesta resultante a la caché dada. Esto es funcionalmente equivalente a llamar fetch(), luego usar put()para agregar los resultados al caché.

[Cache.addAll(requests)](https://developer.mozilla.org/en-US/docs/Web/API/Cache/addAll)

Toma una matriz de URL, las recupera y agrega los objetos de respuesta resultantes a la caché dada.

[Cache.put(request, response)](https://developer.mozilla.org/en-US/docs/Web/API/Cache/put)

Toma tanto una solicitud como su respuesta y la agrega al caché dado.

[Cache.delete(request, options)](https://developer.mozilla.org/en-US/docs/Web/API/Cache/delete)

Encuentra la [Cache](https://developer.mozilla.org/en-US/docs/Web/API/Cache) entrada cuya clave es la solicitud, devolviendo una [Promise](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise" \o "El objeto Promise representa la finalización (o falla) eventual de una operación asincrónica y su valor resultante." \t "_blank)que se resuelve truesi Cachese encuentra y elimina una entrada coincidente . Si no Cachese encuentra ninguna entrada, la promesa se resuelve false.

[Cache.keys(request, options)](https://developer.mozilla.org/en-US/docs/Web/API/Cache/keys)

Devuelve un [Promise](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise" \o "El objeto Promise representa la finalización (o falla) eventual de una operación asincrónica y su valor resultante." \t "_blank)que se resuelve en una matriz de Cacheclaves.