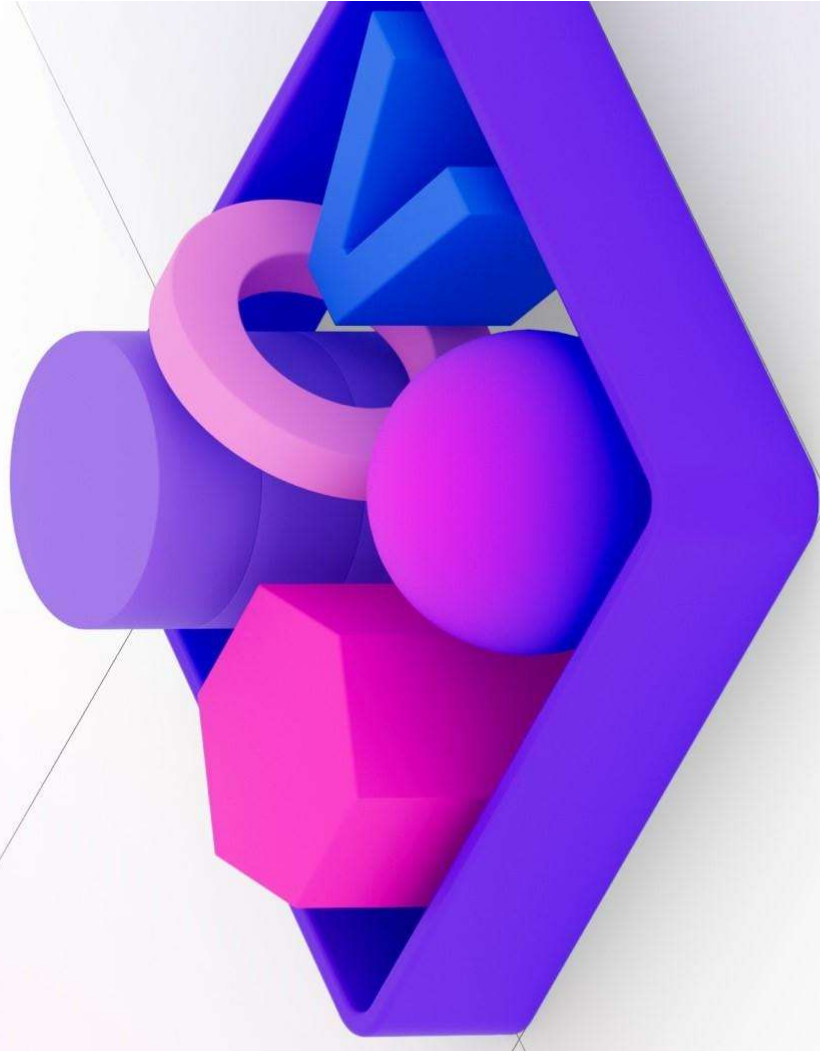


.NET Conf TAIWAN



STUDY4

為 學 習 而 生



徹底搞懂 Blazor 的 PWA 機制

Jimmy Ho

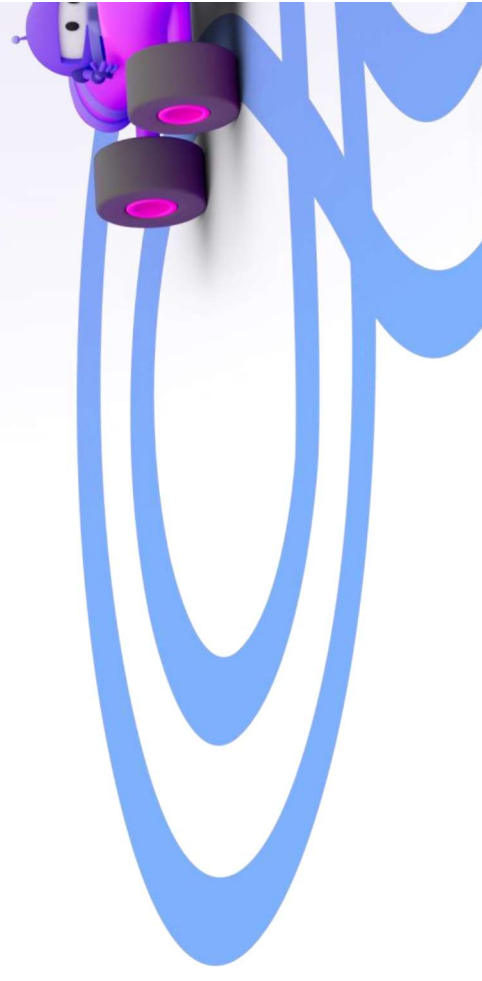


我是誰

- Jimmy
- 任職於資通電腦擔任 ciMes 系統研發
- 熟悉 Blazor/Angular/React 等前端框架
- 演講經驗
 - 2022 Dotnet conf
 - 2021 Dotnet conf
- 部落格：<https://jiaming0708.github.io/>

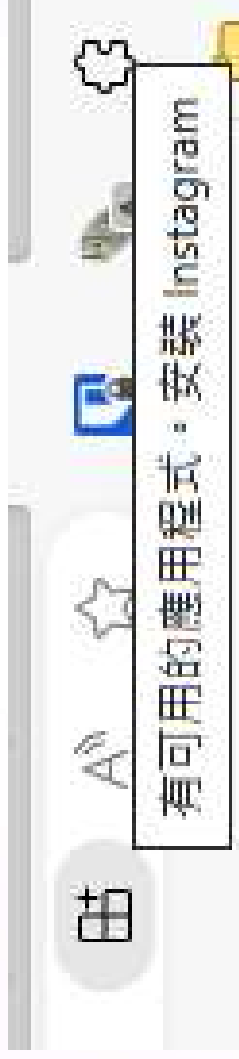


PWA 概念介紹



什麼是 PWA

- 2015 年提出
- **Progressive web app**，漸進式網路應用程式
- 有瀏覽器就可以使用
- 像是 **APP** 一樣可以**安裝**
- 有**版本**的概念 (service worker)
- 可以**離線使用** (service worker)

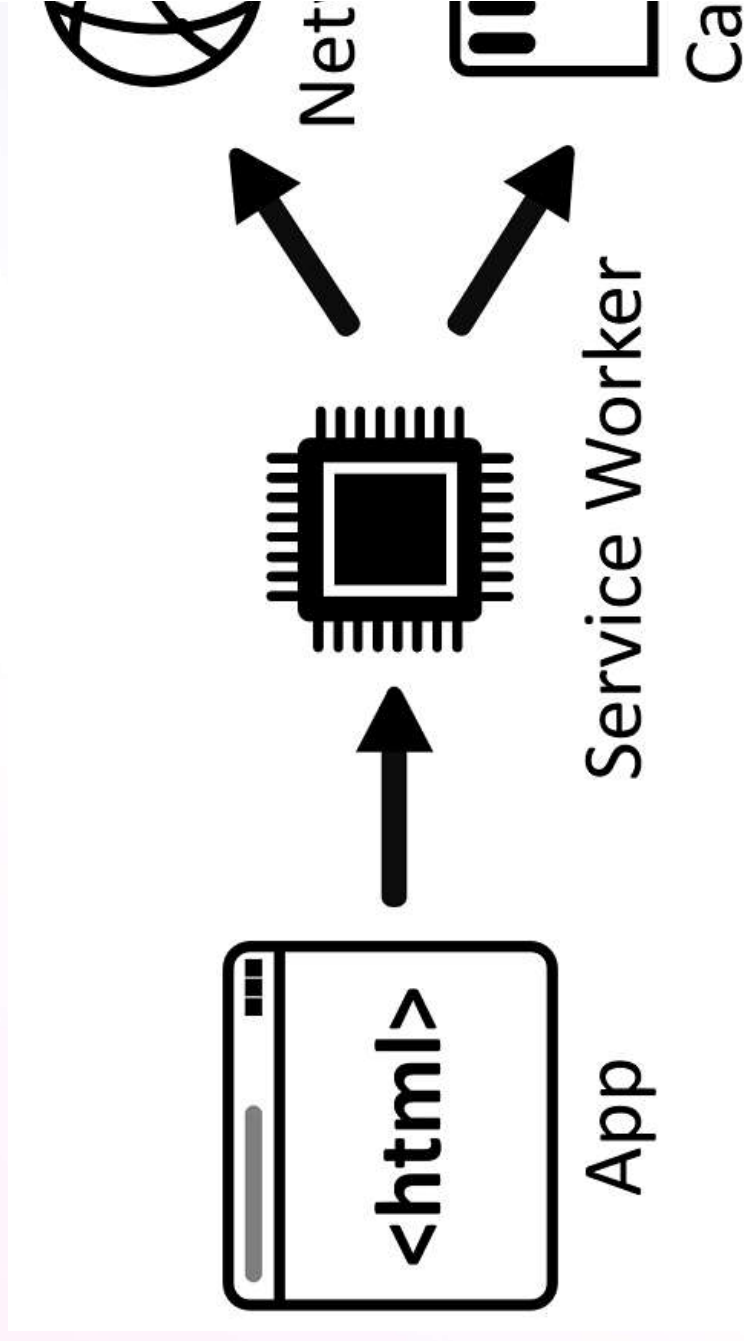


PWA 缺點

- 無法使用部分硬體
- 瀏覽器支援度不一
- 版本控制

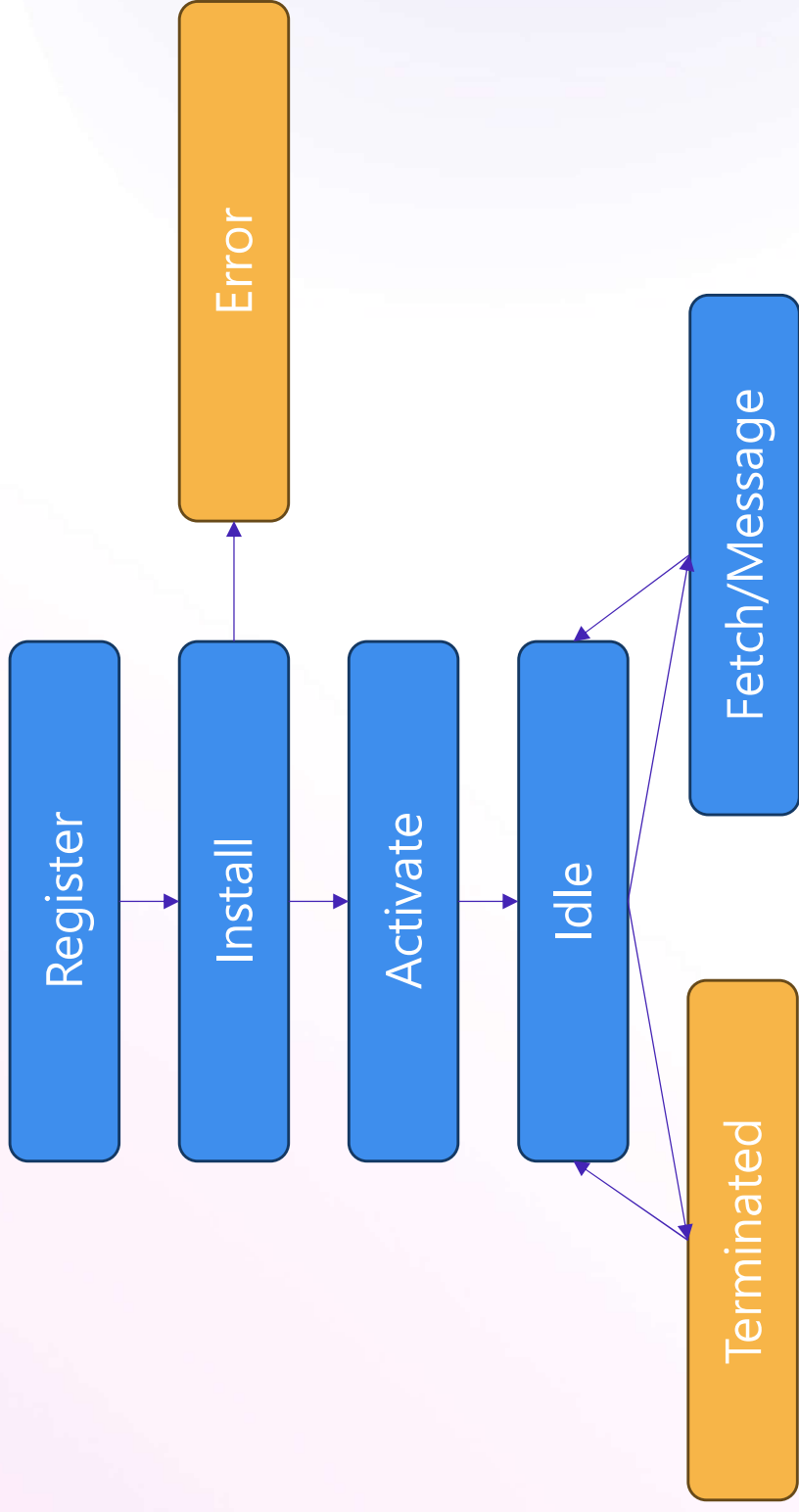
Service Worker

- Web Worker 的另一種應用
- Web 的網路代理
- Cache
- 推播
- 限定 HTTPS 環境



[Use Service Workers to manage network requests - Microsoft Edge Development | Microsoft](#)

Service Worker 生命週期



Register

```
if ('serviceWorker' in navigator) {  
    navigator.serviceWorker.register('service-worker.js');  
}
```

Install

```
self.addEventListener('install', event => {  
  event.waitUntil(  
    // 將檔案放到快取  
    caches  
      .open('mycache-v1')  
      .then(cache => {  
        return cache.add('css/app.css');  
      });  
  });  
});
```

Activate

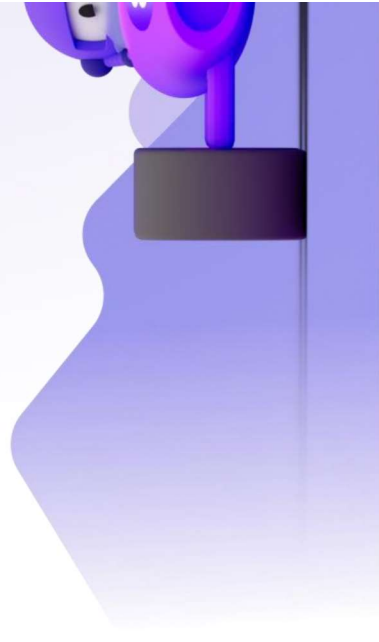
```
self.addEventListener('activate', event => {  
  event.waitUntil(  
    // 刪除舊的快取  
    caches.forEach((cache, cacheName) => {  
      if (cacheName.include('v1')) {  
        return caches.delete(cacheName);  
      }  
    })),  
  );  
});
```

Fetch

```
self.addEventListener("fetch", event => {  
  event.respondWith(async () => {  
    // 嘗試取得 Install 階段所存放的快取  
    const cache = await caches.open('mycache-v1');  
    const cachedResponse = await cache.match(event.request.url);  
  
    return cachedResponse || fetch(event.request.url);  
  });  
});
```

Demo

dotnet new blazorwasm --pwa



cspoj

```
<PropertyGroup>  
  <ServiceWorkerAssetsManifest>service-worker-assets.js  
</ServiceWorkerAssetsManifest>  
</PropertyGroup>  
  
<ItemGroup>  
  <ServiceWorker Include="wwwroot\service-worker.js"  
    PublishedContent="wwwroot\service-worker.published.js"  
  />  
</ItemGroup>
```

index.html

```
<body>  
  <script>navigator.serviceWorker.  
register('service-worker.js');</script>  
</body>
```

兩個 JS 檔案

- service-worker.js
- service-worker.published.js

service-worker.published.js

- 載入 `service-worker-assets.js`
- 定義快取副檔名
- Install
 - 將 `assets.js` 中常見副檔名的檔案做快取
- Activate
 - 移除不是這個版本的快取
- Fetch
 - 導覽的請求並且在 `assets.js` 清單中，則從快取中取得；反之跟 `server`

service-worker-assets.js

- wwwroot 底下(或其他指定路徑)所有檔案路徑及 hash
- 版本
- 編譯後產生

```
self.assetsManifest = {  
  "assets": [  
    {  
      "hash": "sha256-JHedmGrYqlC6lSRAfVAyfvxOVdrRClzapjzpSPg",  
      "url": "index.html"  
    },  
    {  
      "hash": "sha256-1uoJGZA1zW1SycwLPqzQkIJAUF0r5hPav4te25iob",  
      "url": "manifest.webmanifest"  
    },  
    {  
      "hash": "sha256-enKgCMkYmCpfEcmg6AnnbmC40VZVA6aYYSQJZfVr",  
      "url": "sample-data Vweather.json"  
    }  
  ],  
  "version": "07mGFGn0"  
};
```

設定不在 wwwroot 底下的檔案

```
<ItemGroup>  
  <ServiceWorkerAssetsManifestItem Include=  
    "MyDirectory\AnotherFile.json"  
    RelativePath="MyDirectory\AnotherFile.json"  
    AssetUrl="files/AnotherFile.json" />  
</ItemGroup>
```

進階探討



增加判斷 assets.js 中的檔案

```
const shouldServeIndexHtml = event.request.mode === 'navigate'  
    && !manifestUrlList.some(url => url === event.request.url);
```

下載不常見靜態檔案被認為是路由



Hash 不同而下載失敗

Service worker: Install, version offline-cache-dkKmdH5n

✖ Failed to find a valid digest in the 'integrity' attribute for resource '[http://localhost:8080/ framework/blazor.boot.json](http://localhost:8080/framework/blazor.boot.json)' computed SHA-256 integrity 'DwPKHNKsZ8oug3nQubtaiFdLTQ6t7ljRmdbzStJ3nUo='. The resource has been blocked.

✖ Unknown error occurred while trying to verify integrity.

✖ ▶ Uncaught (in promise) TypeError: Failed to fetch
at [service-worker.js:28:54](#)
at async onInstall ([service-worker.js:28:5](#))

[service-wo](#)

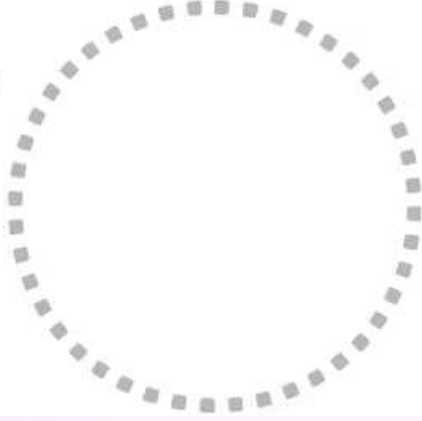
[service-work](#)

移除 hash 檢查

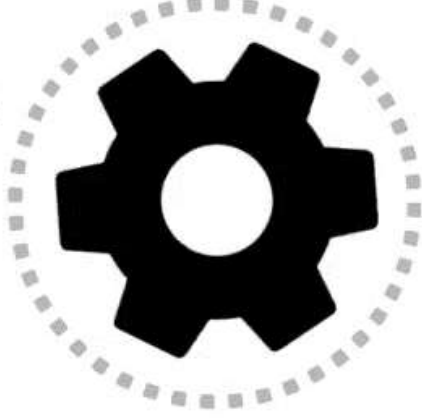
```
.map(asset => new Request(asset.url, {  
  integrity: asset.hash,  
  cache: 'no-cache'  
}))
```


更新機制

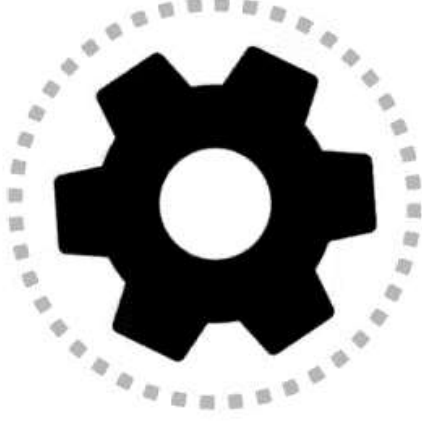
Installing



Waiting



Active



Waiting update

來源

[service-worker.js](#)

收到 2023/12/6 下午 8:20:45

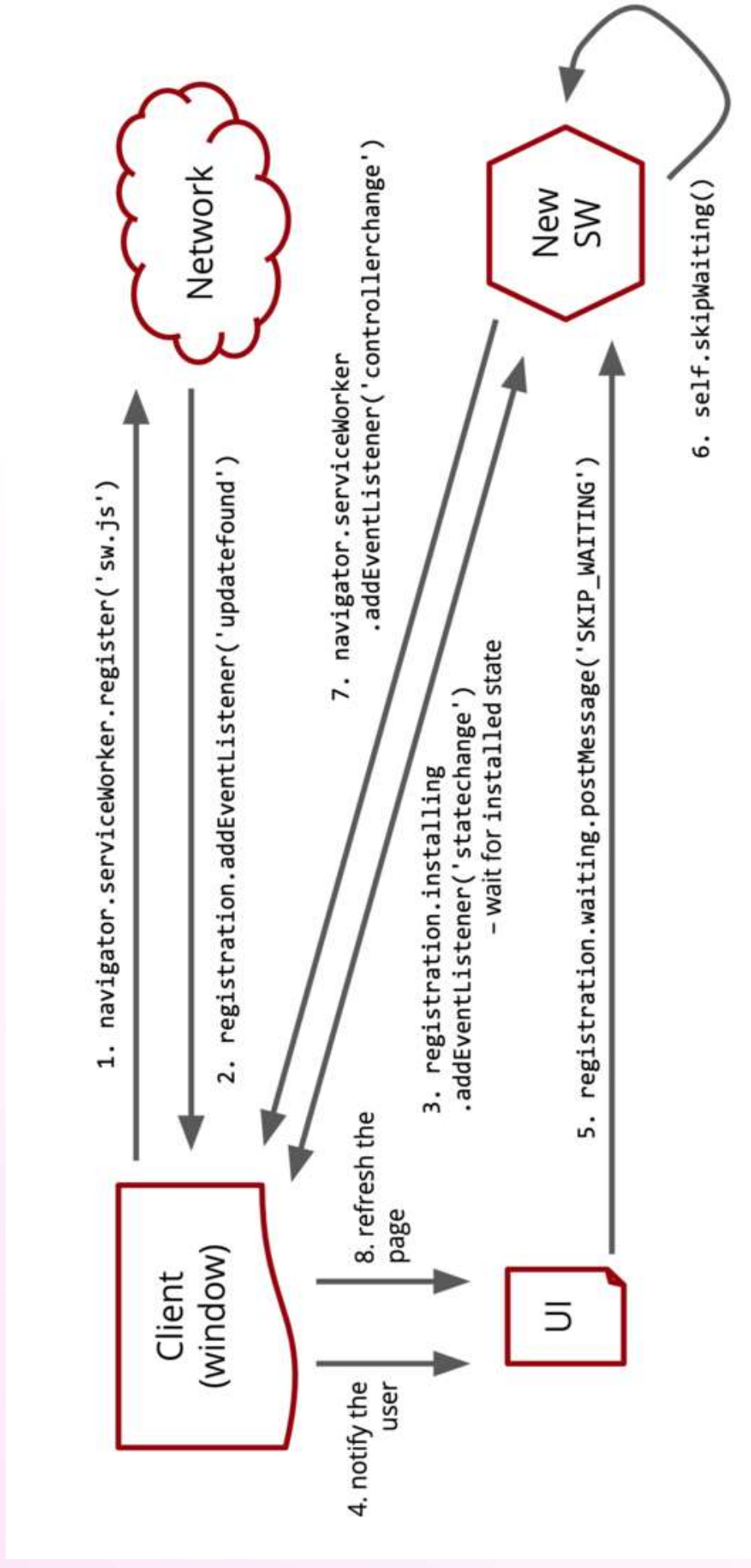
狀態

● #已啟用 2057，且為正在執行 [停止](#)

● #2084 正在等待啟用 [skipWaiting](#)

收到 2023/12/7 下午 5:39:34

如何控制更新



[What Web Can Do Today: Handling Service Worker updates – how to keep the app updated and stay sane](#)

Reference

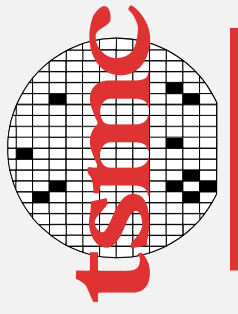
- [A service worker's life - Chrome for Developers](#)
- [Using Service Workers - Web APIs | MDN \(mozilla.org\)](#)
- [ASP.NET Core Blazor Progressive Web Application \(PWA\) | Microsoft Learn](#)
- [What Web Can Do Today: Handling Service Worker updates how to keep the app updated and stay sane](#)

特別感謝

.NET C#
TAI



Microsoft



 Build School

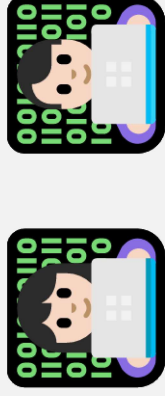
成



HackMD



Microsoft®
Most Valuable
Professional



STUDY4
為 學 習 而 生

Azure Tai
User Group

以及各位參與活動的你們