

API Report File for "@angular/service-worker"

Do not edit this file. It is a report generated by [API Extractor](#).

```
import * as i0 from '@angular/core';
import { ModuleWithProviders } from '@angular/core';
import { Observable } from 'rxjs';

// @public (undocumented)
export class ServiceWorkerModule {
    static register(script: string, opts?: SwRegistrationOptions):
ModuleWithProviders<ServiceWorkerModule>;
    // (undocumented)
    static efac: i0.eeFactoryDeclaration<ServiceWorkerModule, never>;
    // (undocumented)
    static einj: i0.eeInjectorDeclaration<ServiceWorkerModule>;
    // (undocumented)
    static emod: i0.eeNgModuleDeclaration<ServiceWorkerModule, never, never, never>;
}

// @public
export class SwPush {
    constructor(sw: NgswCommChannel);
    get isEnabled(): boolean;
    readonly messages: Observable<object>;
    readonly notificationClicks: Observable<{
        action: string;
        notification: NotificationOptions & {
            title: string;
        };
    }>;
    requestSubscription(options: {
        serverPublicKey: string;
    }): Promise<PushSubscription>;
    readonly subscription: Observable<PushSubscription | null>;
    unsubscribe(): Promise<void>;
    // (undocumented)
    static efac: i0.eeFactoryDeclaration<SwPush, never>;
    // (undocumented)
    static eprov: i0.eeInjectableDeclaration<SwPush>;
}

// @public
export abstract class SwRegistrationOptions {
    enabled?: boolean;
    registrationStrategy?: string | (() => Observable<unknown>);
    scope?: string;
}

// @public
```

```

export class SwUpdate {
  constructor(sw: NgswCommChannel);
  // @deprecated
  readonly activated: Observable<UpdateActivatedEvent>;
  activateUpdate(): Promise<boolean>;
  // @deprecated
  readonly available: Observable<UpdateAvailableEvent>;
  checkForUpdate(): Promise<boolean>;
  get isEnabled(): boolean;
  readonly unrecoverable: Observable<UnrecoverableStateEvent>;
  readonly versionUpdates: Observable<VersionEvent>;
  // (undocumented)
  static efac: i0.ɵɵFactoryDeclaration<SwUpdate, never>;
  // (undocumented)
  static eprov: i0.ɵɵInjectableDeclaration<SwUpdate>;
}

// @public
export interface UnrecoverableStateEvent {
  // (undocumented)
  reason: string;
  // (undocumented)
  type: 'UNRECOVERABLE_STATE';
}

// @public @deprecated
export interface UpdateActivatedEvent {
  // (undocumented)
  current: {
    hash: string;
    appData?: Object;
  };
  // (undocumented)
  previous?: {
    hash: string;
    appData?: Object;
  };
  // (undocumented)
  type: 'UPDATE_ACTIVATED';
}

// @public @deprecated
export interface UpdateAvailableEvent {
  // (undocumented)
  available: {
    hash: string;
    appData?: Object;
  };
  // (undocumented)
  current: {
    hash: string;
    appData?: Object;
  };
}

```

```

    };
    // (undocumented)
    type: 'UPDATE_AVAILABLE';
}

// @public
export interface VersionDetectedEvent {
    // (undocumented)
    type: 'VERSION_DETECTED';
    // (undocumented)
    version: {
        hash: string;
        appData?: object;
    };
}

// @public
export type VersionEvent = VersionDetectedEvent | VersionInstallationFailedEvent |
VersionReadyEvent | NoNewVersionDetectedEvent;

// @public
export interface VersionInstallationFailedEvent {
    // (undocumented)
    error: string;
    // (undocumented)
    type: 'VERSION_INSTALLATION_FAILED';
    // (undocumented)
    version: {
        hash: string;
        appData?: object;
    };
}

// @public
export interface VersionReadyEvent {
    // (undocumented)
    currentVersion: {
        hash: string;
        appData?: object;
    };
    // (undocumented)
    latestVersion: {
        hash: string;
        appData?: object;
    };
    // (undocumented)
    type: 'VERSION_READY';
}

// (No @packageDocumentation comment for this package)

```