API Report File for "@angular/platform-browser"

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

```
import { ComponentRef } from '@angular/core';
import { DebugElement } from '@angular/core';
import { DebugNode } from '@angular/core';
import * as i0 from '@angular/core';
import * as i1 from '@angular/common';
import { InjectionToken } from '@angular/core';
import { ModuleWithProviders } from '@angular/core';
import { NgZone } from '@angular/core';
import { PlatformRef } from '@angular/core';
import { Predicate } from '@angular/core';
import { Sanitizer } from '@angular/core';
import { SecurityContext } from '@angular/core';
import { StaticProvider } from '@angular/core';
import { Type } from '@angular/core';
import { Version } from '@angular/core';

// @public
export class BrowserModule {
    constructor(parentModule: BrowserModule | null);
    static withServerTransition(params: {
        appId: string;
    }): ModuleWithProviders<BrowserModule>;
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<BrowserModule, [{ optional: true; skipSelf:
true; }]>;
    // (undocumented)
    static ɵinj: i0.ɵɵInjectorDeclaration<BrowserModule>;
    // (undocumented)
    static ɵmod: i0.ɵɵNgModuleDeclaration<BrowserModule, never, never, [typeof
i1.CommonModule, typeof i0.ApplicationModule]>;
}

// @public
export class BrowserTransferStateModule {
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<BrowserTransferStateModule, never>;
    // (undocumented)
    static ɵinj: i0.ɵɵInjectorDeclaration<BrowserTransferStateModule>;
    // (undocumented)
    static ɵmod: i0.ɵɵNgModuleDeclaration<BrowserTransferStateModule, never, never,
never>;
}

// @public
export class By {
    static all(): Predicate<DebugNode>;
```

```
    static css(selector: string): Predicate<DebugElement>;
    static directive(type: Type<any>): Predicate<DebugNode>;
}

// @public
export function disableDebugTools(): void;

// @public
export abstract class DomSanitizer implements Sanitizer {
    abstract bypassSecurityTrustHtml(value: string): SafeHtml;
    abstract bypassSecurityTrustResourceUrl(value: string): SafeResourceUrl;
    abstract bypassSecurityTrustScript(value: string): SafeScript;
    abstract bypassSecurityTrustStyle(value: string): SafeStyle;
    abstract bypassSecurityTrustUrl(value: string): SafeUrl;
    abstract sanitize(context: SecurityContext, value: SafeValue | string | null):
string | null;
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<DomSanitizer, never>;
    // (undocumented)
    static ɵprov: i0.ɵɵInjectableDeclaration<DomSanitizer>;
}

// @public
export function enableDebugTools<T>(ref: ComponentRef<T>): ComponentRef<T>;

// @public
export const EVENT_MANAGER_PLUGINS: InjectionToken<EventManagerPlugin[]>;

// @public
export class EventManager {
    constructor(plugins: EventManagerPlugin[], _zone: NgZone);
    addEventListener(element: HTMLElement, eventName: string, handler: Function):
Function;
    // @deprecated
    addGlobalEventListener(target: string, eventName: string, handler: Function):
Function;
    getZone(): NgZone;
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<EventManager, never>;
    // (undocumented)
    static ɵprov: i0.ɵɵInjectableDeclaration<EventManager>;
}

// @public
export const HAMMER_GESTURE_CONFIG: InjectionToken<HammerGestureConfig>;

// @public
export const HAMMER_LOADER: InjectionToken<HammerLoader>;

// @public
export class HammerGestureConfig {
    buildHammer(element: HTMLElement): HammerInstance;
```

```
    events: string[];
    options?: {
        cssProps?: any;
        domEvents?: boolean;
        enable?: boolean | ((manager: any) => boolean);
        preset?: any[];
        touchAction?: string;
        recognizers?: any[];
        inputClass?: any;
        inputTarget?: EventTarget;
    };
    overrides: {
        [key: string]: Object;
    };
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<HammerGestureConfig, never>;
    // (undocumented)
    static ɵprov: i0.ɵɵInjectableDeclaration<HammerGestureConfig>;
}

// @public
export type HammerLoader = () => Promise<void>;

// @public
export class HammerModule {
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<HammerModule, never>;
    // (undocumented)
    static ɵinj: i0.ɵɵInjectorDeclaration<HammerModule>;
    // (undocumented)
    static ɵmod: i0.ɵɵNgModuleDeclaration<HammerModule, never, never, never>;
}

// @public
export function makeStateKey<T = void>(key: string): StateKey<T>;

// @public
export class Meta {
    constructor(_doc: any);
    addTag(tag: MetaDefinition, forceCreation?: boolean): HTMLMetaElement | null;
    addTags(tags: MetaDefinition[], forceCreation?: boolean): HTMLMetaElement[];
    getTag(attrSelector: string): HTMLMetaElement | null;
    getTags(attrSelector: string): HTMLMetaElement[];
    removeTag(attrSelector: string): void;
    removeTagElement(meta: HTMLMetaElement): void;
    updateTag(tag: MetaDefinition, selector?: string): HTMLMetaElement | null;
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<Meta, never>;
    // (undocumented)
    static ɵprov: i0.ɵɵInjectableDeclaration<Meta>;
}
```

```
// @public
export type MetaDefinition = {
    charset?: string;
    content?: string;
    httpEquiv?: string;
    id?: string;
    itemprop?: string;
    name?: string;
    property?: string;
    scheme?: string;
    url?: string;
} & {
    [prop: string]: string;
};

// @public
export const platformBrowser: (extraProviders?: StaticProvider[]) => PlatformRef;

// @public
export interface SafeHtml extends SafeValue {
}

// @public
export interface SafeResourceUrl extends SafeValue {
}

// @public
export interface SafeScript extends SafeValue {
}

// @public
export interface SafeStyle extends SafeValue {
}

// @public
export interface SafeUrl extends SafeValue {
}

// @public
export interface SafeValue {
}

// @public
export type StateKey<T> = string & {
    __not_a_string: never;
    __value_type?: T;
};

// @public
export class Title {
    constructor(_doc: any);
    getTitle(): string;
```

```
    setTitle(newTitle: string): void;
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<Title, never>;
    // (undocumented)
    static ɵprov: i0.ɵɵInjectableDeclaration<Title>;
}

// @public
export class TransferState {
    get<T>(key: StateKey<T>, defaultValue: T): T;
    hasKey<T>(key: StateKey<T>): boolean;
    onSerialize<T>(key: StateKey<T>, callback: () => T): void;
    remove<T>(key: StateKey<T>): void;
    set<T>(key: StateKey<T>, value: T): void;
    toJson(): string;
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<TransferState, never>;
    // (undocumented)
    static ɵprov: i0.ɵɵInjectableDeclaration<TransferState>;
}

// @public (undocumented)
export const VERSION: Version;

// (No @packageDocumentation comment for this package)
```