API Report File for "@angular/common_testing"

> *Do not edit this file. It is a report generated by [API Extractor](https://api-extractor.com/).*

```
import * as i0 from '@angular/core';
import { InjectionToken } from '@angular/core';
import { Location as Location_2 } from '@angular/common';
import { LocationChangeListener } from '@angular/common';
import { LocationStrategy } from '@angular/common';
import { PlatformLocation } from '@angular/common';
import { SubscriptionLike } from 'rxjs';

// @public
export const MOCK_PLATFORM_LOCATION_CONFIG:
InjectionToken<MockPlatformLocationConfig>;

// @public
export class MockLocationStrategy extends LocationStrategy {
    constructor();
    // (undocumented)
    back(): void;
    // (undocumented)
    forward(): void;
    // (undocumented)
    getBaseHref(): string;
    // (undocumented)
    getState(): unknown;
    // (undocumented)
    internalBaseHref: string;
    // (undocumented)
    internalPath: string;
    // (undocumented)
    internalTitle: string;
    // (undocumented)
    onPopState(fn: (value: any) => void): void;
    // (undocumented)
    path(includeHash?: boolean): string;
    // (undocumented)
    prepareExternalUrl(internal: string): string;
    // (undocumented)
    pushState(ctx: any, title: string, path: string, query: string): void;
    // (undocumented)
    replaceState(ctx: any, title: string, path: string, query: string): void;
    // (undocumented)
    simulatePopState(url: string): void;
    // (undocumented)
    urlChanges: string[];
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<MockLocationStrategy, never>;
    // (undocumented)
    static ɵprov: i0.ɵɵInjectableDeclaration<MockLocationStrategy>;
}

// @public
export class MockPlatformLocation implements PlatformLocation {
    constructor(config?: MockPlatformLocationConfig);
    // (undocumented)
    back(): void;
    // (undocumented)
    forward(): void;
    // (undocumented)
    getBaseHrefFromDOM(): string;
    // (undocumented)
    getState(): unknown;
    // (undocumented)
    get hash(): string;
    // (undocumented)
    historyGo(relativePosition?: number): void;
    // (undocumented)
    get hostname(): string;
    // (undocumented)
    get href(): string;
    // (undocumented)
    onHashChange(fn: LocationChangeListener): VoidFunction;
    // (undocumented)
    onPopState(fn: LocationChangeListener): VoidFunction;
    // (undocumented)
    get pathname(): string;
    // (undocumented)
    get port(): string;
    // (undocumented)
    get protocol(): string;
    // (undocumented)
    pushState(state: any, title: string, newUrl: string): void;
    // (undocumented)
    replaceState(state: any, title: string, newUrl: string): void;
    // (undocumented)
    get search(): string;
    // (undocumented)
    get state(): unknown;
    // (undocumented)
    get url(): string;
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<MockPlatformLocation, [{ optional: true;
}]>;
    // (undocumented)
    static ɵprov: i0.ɵɵInjectableDeclaration<MockPlatformLocation>;
}

// @public
export interface MockPlatformLocationConfig {
    // (undocumented)
    appBaseHref?: string;
    // (undocumented)
    startUrl?: string;
}

// @public
export class SpyLocation implements Location_2 {
    // (undocumented)
    back(): void;
    // (undocumented)
    forward(): void;
    // (undocumented)
    getState(): unknown;
    // (undocumented)
    go(path: string, query?: string, state?: any): void;
    // (undocumented)
    historyGo(relativePosition?: number): void;
    // (undocumented)
    isCurrentPathEqualTo(path: string, query?: string): boolean;
    // (undocumented)
    ngOnDestroy(): void;
    // (undocumented)
    normalize(url: string): string;
    // (undocumented)
    onUrlChange(fn: (url: string, state: unknown) => void): VoidFunction;
    // (undocumented)
    path(): string;
    // (undocumented)
    prepareExternalUrl(url: string): string;
    // (undocumented)
    replaceState(path: string, query?: string, state?: any): void;
    // (undocumented)
    setBaseHref(url: string): void;
    // (undocumented)
    setInitialPath(url: string): void;
    // (undocumented)
    simulateHashChange(pathname: string): void;
    // (undocumented)
    simulateUrlPop(pathname: string): void;
    // (undocumented)
    subscribe(onNext: (value: any) => void, onThrow?: ((error: any) => void) | null,
onReturn?: (() => void) | null): SubscriptionLike;
    // (undocumented)
    urlChanges: string[];
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<SpyLocation, never>;
    // (undocumented)
    static ɵprov: i0.ɵɵInjectableDeclaration<SpyLocation>;
}

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