API Report File for “@angular/core_testing”

Do not edit this file. It is a report generated by API Extractor.

```
import { ChangeDetectorRef } from '@angular/core';
import { Component } from '@angular/core';
import { ComponentRef } from '@angular/core';
import { DebugElement } from '@angular/core';
import { Directive } from '@angular/core';
import { ElementRef } from '@angular/core';
import { InjectFlags } from '@angular/core';
import { InjectionToken } from '@angular/core';
import { NgModule } from '@angular/core';
import { NgZone } from '@angular/core';
import { Pipe } from '@angular/core';
import { PlatformRef } from '@angular/core';
import { ProviderToken } from '@angular/core';
import { SchemaMetadata } from '@angular/core';
import { Type } from '@angular/core';

// @public
export const __core_private_testing_placeholder__ = "";

// @public @deprecated (undocumented)
export function async(fn: Function): (done: any) => any;

// @public
export class ComponentFixture<T> {
    constructor(componentRef: ComponentRef<T>, ngZone: NgZone | null, _autoDetect: boolean)
    autoDetectChanges(autoDetect?: boolean): void;
    changeDetectorRef: ChangeDetectorRef;
    checkNoChanges(): void;
    componentInstance: T;
    // (undocumented)
    componentRef: ComponentRef<T>;
    debugElement: DebugElement;
    destroy(): void;
    detectChanges(checkNoChanges?: boolean): void;
    elementRef: ElementRef;
    isStable(): boolean;
    nativeElement: any;
    // (undocumented)
    ngZone: NgZone | null;
    whenRenderingDone(): Promise<any>;
    whenStable(): Promise<any>;
}

// @public (undocumented)
export const ComponentFixtureAutoDetect: InjectionToken<boolean[]>;

// @public (undocumented)
export const ComponentFixtureNoNgZone: InjectionToken<boolean[]>;

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

// @public
export function fakeAsync(fn: Function): (...args: any[]) => any;

// @public
export function flush(maxTurns?: number): number;

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

// @public
export const getTestBed: () => TestBed;

// @public
export function inject(tokens: any[], fn: Function): () => any;

// @public (undocumented)
export class InjectSetupWrapper {
    constructor(_moduleDef: () => TestModuleMetadata);
    // (undocumented)
    inject(tokens: any[], fn: Function): () => any;
}

// @public
export type MetadataOverride<T> = {
    add?: Partial<T>;
    remove?: Partial<T>;
    set?: Partial<T>;
};

// @public
export interface ModuleTeardownOptions {
    destroyAfterEach: boolean;
    rethrowErrors?: boolean;
}
```

```
// @public
export function resetFakeAsyncZone(): void;

// @public (undocumented)
export interface TestBed {
    // (undocumented)
    compileComponents(): Promise<any>;
    // (undocumented)
    configureCompiler(config: {
        providers?: any[];
        useJit?: boolean;
    }): void;
    // (undocumented)
    configureTestingModule(moduleDef: TestModuleMetadata): void;
    // (undocumented)
    createComponent<T>(component: Type<T>): ComponentFixture<T>;
    // (undocumented)
    execute(tokens: any[], fn: Function, context?: any): any;
    // @deprecated (undocumented)
    get<T>(token: ProviderToken<T>, notFoundValue?: T, flags?: InjectFlags): any;
    // @deprecated (undocumented)
    get(token: any, notFoundValue?: any): any;
    initTestEnvironment(ngModule: Type<any> | Type<any>[], platform: PlatformRef, options?:
    // @deprecated
    initTestEnvironment(ngModule: Type<any> | Type<any>[], platform: PlatformRef, aotSummari
    // (undocumented)
    inject<T>(token: ProviderToken<T>, notFoundValue?: T, flags?: InjectFlags): T;
    // (undocumented)
    inject<T>(token: ProviderToken<T>, notFoundValue: null, flags?: InjectFlags): T | null;
    // (undocumented)
    ngModule: Type<any> | Type<any>[];
    // (undocumented)
    overrideComponent(component: Type<any>, override: MetadataOverride<Component>): void;
    // (undocumented)
    overrideDirective(directive: Type<any>, override: MetadataOverride<Directive>): void;
    // (undocumented)
    overrideModule(ngModule: Type<any>, override: MetadataOverride<NgModule>): void;
    // (undocumented)
    overridePipe(pipe: Type<any>, override: MetadataOverride<Pipe>): void;
    overrideProvider(token: any, provider: {
        useFactory: Function;
        deps: any[];
    }): void;
    // (undocumented)
    overrideProvider(token: any, provider: {
        useValue: any;
    }): void;
    // (undocumented)
    overrideProvider(token: any, provider: {
        useFactory?: Function;
        useValue?: any;
        deps?: any[];
    }): void;
    // (undocumented)
    overrideTemplateUsingTestingModule(component: Type<any>, template: string): void;
    // (undocumented)
    platform: PlatformRef;
    resetTestEnvironment(): void;
    // (undocumented)
    resetTestingModule(): void;
}

// @public
export const TestBed: TestBedStatic;

// @public
export interface TestBedStatic {
    // (undocumented)
    new (...args: any[]): TestBed;
    compileComponents(): Promise<any>;
    configureCompiler(config: {
        providers?: any[];
        useJit?: boolean;
    }): TestBedStatic;
    configureTestingModule(moduleDef: TestModuleMetadata): TestBedStatic;
    // (undocumented)
    createComponent<T>(component: Type<T>): ComponentFixture<T>;
    // @deprecated (undocumented)
    get<T>(token: ProviderToken<T>, notFoundValue?: T, flags?: InjectFlags): any;
    // @deprecated (undocumented)
    get(token: any, notFoundValue?: any): any;
    initTestEnvironment(ngModule: Type<any> | Type<any>[], platform: PlatformRef, options?:
    // @deprecated
    initTestEnvironment(ngModule: Type<any> | Type<any>[], platform: PlatformRef, aotSummari
    // (undocumented)
    inject<T>(token: ProviderToken<T>, notFoundValue?: T, flags?: InjectFlags): T;
    // (undocumented)
    inject<T>(token: ProviderToken<T>, notFoundValue: null, flags?: InjectFlags): T | null;
    // (undocumented)
    overrideComponent(component: Type<any>, override: MetadataOverride<Component>): TestBedS
    // (undocumented)
    overrideDirective(directive: Type<any>, override: MetadataOverride<Directive>): TestBedS
    // (undocumented)
    overrideModule(ngModule: Type<any>, override: MetadataOverride<NgModule>): TestBedStatic
    // (undocumented)
    overridePipe(pipe: Type<any>, override: MetadataOverride<Pipe>): TestBedStatic;
    overrideProvider(token: any, provider: {
        useFactory: Function;
        deps: any[];
    }): TestBedStatic;
    // (undocumented)
    overrideProvider(token: any, provider: {
        useValue: any;
    }): TestBedStatic;
    // (undocumented)
    overrideProvider(token: any, provider: {
        useFactory?: Function;
        useValue?: any;
        deps?: any[];
    }): TestBedStatic;
    // (undocumented)
    overrideTemplate(component: Type<any>, template: string): TestBedStatic;
    overrideTemplateUsingTestingModule(component: Type<any>, template: string): TestBedStati
    resetTestEnvironment(): void;
    // (undocumented)
    resetTestingModule(): TestBedStatic;
}

// @public
export class TestComponentRenderer {
    // (undocumented)
    insertRootElement(rootElementId: string): void;
    // (undocumented)
    removeAllRootElements?(): void;
}

// @public (undocumented)
export interface TestEnvironmentOptions {
    // @deprecated
    aotSummaries?: () => any[];
    teardown?: ModuleTeardownOptions;
}

// @public (undocumented)
export type TestModuleMetadata = {
    providers?: any[];
    declarations?: any[];
    imports?: any[];
    schemas?: Array<SchemaMetadata | any[]>;
    aotSummaries?: () => any[];
    teardown?: ModuleTeardownOptions;
};

// @public
export function tick(millis?: number, tickOptions?: {
    processNewMacroTasksSynchronously: boolean;
}): void;

// @public
export function waitForAsync(fn: Function): (done: any) => any;

// @public (undocumented)
export function withModule(moduleDef: TestModuleMetadata): InjectSetupWrapper;

// @public (undocumented)
export function withModule(moduleDef: TestModuleMetadata, fn: Function): () => any;

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