

API Report File for "@angular/common"

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

```
import { ChangeDetectorRef } from '@angular/core';
import { DoCheck } from '@angular/core';
import { ElementRef } from '@angular/core';
import * as i0 from '@angular/core';
import { InjectionToken } from '@angular/core';
import { Injector } from '@angular/core';
import { IterableDiffers } from '@angular/core';
import { KeyValueDiffers } from '@angular/core';
import { NgIterable } from '@angular/core';
import { NgModuleFactory } from '@angular/core';
import { Observable } from 'rxjs';
import { OnChanges } from '@angular/core';
import { OnDestroy } from '@angular/core';
import { PipeTransform } from '@angular/core';
import { Renderer2 } from '@angular/core';
import { SimpleChanges } from '@angular/core';
import { Subscribable } from 'rxjs';
import { SubscriptionLike } from 'rxjs';
import { TemplateRef } from '@angular/core';
import { TrackByFunction } from '@angular/core';
import { Type } from '@angular/core';
import { Version } from '@angular/core';
import { ViewContainerRef } from '@angular/core';

// @public
export const APP_BASE_HREF: InjectionToken<string>;

// @public
export class AsyncPipe implements OnDestroy, PipeTransform {
  constructor(_ref: ChangeDetectorRef);
  // (undocumented)
  ngOnDestroy(): void;
  // (undocumented)
  transform<T>(obj: Observable<T> | Subscribable<T> | Promise<T>): T | null;
  // (undocumented)
  transform<T>(obj: null | undefined): null;
  // (undocumented)
  transform<T>(obj: Observable<T> | Subscribable<T> | Promise<T> | null |
undefined): T | null;
  // (undocumented)
  static efac: i0.ɵɵFactoryDeclaration<AsyncPipe, never>;
  // (undocumented)
  static epipe: i0.ɵɵPipeDeclaration<AsyncPipe, "async">;
}

// @public
```

```

export class CommonModule {
  // (undocumented)
  static efac: i0.eeFactoryDeclaration<CommonModule, never>;
  // (undocumented)
  static einj: i0.eeInjectorDeclaration<CommonModule>;
  // (undocumented)
  static emod: i0.eeNgModuleDeclaration<CommonModule, [typeof i1.NgClass, typeof
i2.NgComponentOutlet, typeof i3.NgForOf, typeof i4.NgIf, typeof i5.NgTemplateOutlet,
typeof i6.NgStyle, typeof i7.NgSwitch, typeof i7.NgSwitchCase, typeof
i7.NgSwitchDefault, typeof i8.NgPlural, typeof i8.NgPluralCase, typeof i9.AsyncPipe,
typeof i10.UpperCasePipe, typeof i10.LowerCasePipe, typeof i11.JsonPipe, typeof
i12.SlicePipe, typeof i13.DecimalPipe, typeof i13.PercentPipe, typeof
i10.TitleCasePipe, typeof i13.CurrencyPipe, typeof i14.DatePipe, typeof
i15.I18nPluralPipe, typeof i16.I18nSelectPipe, typeof i17.KeyValuePipe], never,
[typeof i1.NgClass, typeof i2.NgComponentOutlet, typeof i3.NgForOf, typeof i4.NgIf,
typeof i5.NgTemplateOutlet, typeof i6.NgStyle, typeof i7.NgSwitch, typeof
i7.NgSwitchCase, typeof i7.NgSwitchDefault, typeof i8.NgPlural, typeof
i8.NgPluralCase, typeof i9.AsyncPipe, typeof i10.UpperCasePipe, typeof
i10.LowerCasePipe, typeof i11.JsonPipe, typeof i12.SlicePipe, typeof
i13.DecimalPipe, typeof i13.PercentPipe, typeof i10.TitleCasePipe, typeof
i13.CurrencyPipe, typeof i14.DatePipe, typeof i15.I18nPluralPipe, typeof
i16.I18nSelectPipe, typeof i17.KeyValuePipe]>;
}

// @public
export class CurrencyPipe implements PipeTransform {
  constructor(_locale: string, _defaultCurrencyCode?: string);
  // (undocumented)
  transform(value: number | string, currencyCode?: string, display?: 'code' |
'symbol' | 'symbol-narrow' | string | boolean, digitsInfo?: string, locale?:
string): string | null;
  // (undocumented)
  transform(value: null | undefined, currencyCode?: string, display?: 'code' |
'symbol' | 'symbol-narrow' | string | boolean, digitsInfo?: string, locale?:
string): null;
  // (undocumented)
  transform(value: number | string | null | undefined, currencyCode?: string,
display?: 'code' | 'symbol' | 'symbol-narrow' | string | boolean, digitsInfo?:
string, locale?: string): string | null;
  // (undocumented)
  static efac: i0.eeFactoryDeclaration<CurrencyPipe, never>;
  // (undocumented)
  static epipe: i0.eePipeDeclaration<CurrencyPipe, "currency">;
}

// @public
export const DATE_PIPE_DEFAULT_TIMEZONE: InjectionToken<string>;

// @public
export class DatePipe implements PipeTransform {
  constructor(locale: string, defaultTimezone?: string | null | undefined);
  // (undocumented)

```

```

        transform(value: Date | string | number, format?: string, timezone?: string,
locale?: string): string | null;
        // (undocumented)
        transform(value: null | undefined, format?: string, timezone?: string, locale?:
string): null;
        // (undocumented)
        transform(value: Date | string | number | null | undefined, format?: string,
timezone?: string, locale?: string): string | null;
        // (undocumented)
        static efac: i0.eeFactoryDeclaration<DatePipe, [null, { optional: true; }]>;
        // (undocumented)
        static epipe: i0.eePipeDeclaration<DatePipe, "date">;
    }

// @public
export class DecimalPipe implements PipeTransform {
    constructor(_locale: string);
    // (undocumented)
    transform(value: number | string, digitsInfo?: string, locale?: string): string
| null;
    // (undocumented)
    transform(value: null | undefined, digitsInfo?: string, locale?: string): null;
    // (undocumented)
    transform(value: number | string | null | undefined, digitsInfo?: string,
locale?: string): string | null;
    // (undocumented)
    static efac: i0.eeFactoryDeclaration<DecimalPipe, never>;
    // (undocumented)
    static epipe: i0.eePipeDeclaration<DecimalPipe, "number">;
}

// @public
export const DOCUMENT: InjectionToken<Document>;

// @public
export function formatCurrency(value: number, locale: string, currency: string,
currencyCode?: string, digitsInfo?: string): string;

// @public
export function formatDate(value: string | number | Date, format: string, locale:
string, timezone?: string): string;

// @public
export function formatNumber(value: number, locale: string, digitsInfo?: string):
string;

// @public
export function formatPercent(value: number, locale: string, digitsInfo?: string):
string;

// @public
export enum FormatWidth {

```

```

    Full = 3,
    Long = 2,
    Medium = 1,
    Short = 0
}

// @public
export enum FormStyle {
    // (undocumented)
    Format = 0,
    // (undocumented)
    Standalone = 1
}

// @public
export function getCurrencySymbol(code: string, format: 'wide' | 'narrow', locale?:
string): string;

// @public
export function getLocaleCurrencyCode(locale: string): string | null;

// @public
export function getLocaleCurrencyName(locale: string): string | null;

// @public
export function getLocaleCurrencySymbol(locale: string): string | null;

// @public
export function getLocaleDateFormat(locale: string, width: FormatWidth): string;

// @public
export function getLocaleDateTimeFormat(locale: string, width: FormatWidth): string;

// @public
export function getLocaleDayNames(locale: string, formStyle: FormStyle, width:
TranslationWidth): ReadonlyArray<string>;

// @public
export function getLocaleDayPeriods(locale: string, formStyle: FormStyle, width:
TranslationWidth): Readonly<[string, string]>;

// @public
export function getLocaleDirection(locale: string): 'ltr' | 'rtl';

// @public
export function getLocaleEraNames(locale: string, width: TranslationWidth):
Readonly<[string, string]>;

// @public
export function getLocaleExtraDayPeriodRules(locale: string): (Time | [Time, Time])
[];

```

```

// @public
export function getLocaleExtraDayPeriods(locale: string, formStyle: FormStyle,
width: TranslationWidth): string[];

// @public
export function getLocaleFirstDayOfWeek(locale: string): WeekDay;

// @public
export function getLocaleId(locale: string): string;

// @public
export function getLocaleMonthNames(locale: string, formStyle: FormStyle, width:
TranslationWidth): ReadonlyArray<string>;

// @public
export function getLocaleNumberFormat(locale: string, type: NumberFormatStyle):
string;

// @public
export function getLocaleNumberSymbol(locale: string, symbol: NumberSymbol): string;

// @public
export const getLocalePluralCase: (locale: string) => ((value: number) => Plural);

// @public
export function getLocaleTimeFormat(locale: string, width: FormatWidth): string;

// @public
export function getLocaleWeekEndRange(locale: string): [WeekDay, WeekDay];

// @public
export function getNumberOfCurrencyDigits(code: string): number;

// @public
export class HashLocationStrategy extends LocationStrategy implements OnDestroy {
  constructor(_platformLocation: PlatformLocation, _baseHref?: string);
  // (undocumented)
  back(): void;
  // (undocumented)
  forward(): void;
  // (undocumented)
  getBaseHref(): string;
  // (undocumented)
  historyGo(relativePosition?: number): void;
  // (undocumented)
  ngOnDestroy(): void;
  // (undocumented)
  onPopState(fn: LocationChangeListener): void;
  // (undocumented)
  path(includeHash?: boolean): string;
  // (undocumented)
  prepareExternalUrl(internal: string): string;

```

```

    // (undocumented)
    pushState(state: any, title: string, path: string, queryParams: string): void;
    // (undocumented)
    replaceState(state: any, title: string, path: string, queryParams: string):
void;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<HashLocationStrategy, [null, { optional:
true; }]>;
    // (undocumented)
    static eprov: i0.ɵɵInjectableDeclaration<HashLocationStrategy>;
}

// @public
export class I18nPluralPipe implements PipeTransform {
    constructor(_localization: NgLocalization);
    // (undocumented)
    transform(value: number | null | undefined, pluralMap: {
        [count: string]: string;
    }, locale?: string): string;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<I18nPluralPipe, never>;
    // (undocumented)
    static epipe: i0.ɵɵPipeDeclaration<I18nPluralPipe, "i18nPlural">;
}

// @public
export class I18nSelectPipe implements PipeTransform {
    // (undocumented)
    transform(value: string | null | undefined, mapping: {
        [key: string]: string;
    }): string;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<I18nSelectPipe, never>;
    // (undocumented)
    static epipe: i0.ɵɵPipeDeclaration<I18nSelectPipe, "i18nSelect">;
}

// @public
export function isPlatformBrowser(platformId: Object): boolean;

// @public
export function isPlatformServer(platformId: Object): boolean;

// @public
export function isPlatformWorkerApp(platformId: Object): boolean;

// @public
export function isPlatformWorkerUi(platformId: Object): boolean;

// @public
export class JsonPipe implements PipeTransform {
    // (undocumented)

```

```

    transform(value: any): string;
    // (undocumented)
    static efac: i0.eeFactoryDeclaration<JsonPipe, never>;
    // (undocumented)
    static epipe: i0.eePipeDeclaration<JsonPipe, "json">;
}

// @public
export interface KeyValue<K, V> {
    // (undocumented)
    key: K;
    // (undocumented)
    value: V;
}

// @public
export class KeyValuePipe implements PipeTransform {
    constructor(differs: KeyValueDiffers);
    // (undocumented)
    transform<K, V>(input: ReadonlyMap<K, V>, compareFn?: (a: KeyValue<K, V>, b:
    KeyValue<K, V>) => number): Array<KeyValue<K, V>>;
    // (undocumented)
    transform<K extends number, V>(input: Record<K, V>, compareFn?: (a:
    KeyValue<string, V>, b: KeyValue<string, V>) => number): Array<KeyValue<string, V>>;
    // (undocumented)
    transform<K extends string, V>(input: Record<K, V> | ReadonlyMap<K, V>,
    compareFn?: (a: KeyValue<K, V>, b: KeyValue<K, V>) => number): Array<KeyValue<K,
    V>>;
    // (undocumented)
    transform(input: null | undefined, compareFn?: (a: KeyValue<unknown, unknown>,
    b: KeyValue<unknown, unknown>) => number): null;
    // (undocumented)
    transform<K, V>(input: ReadonlyMap<K, V> | null | undefined, compareFn?: (a:
    KeyValue<K, V>, b: KeyValue<K, V>) => number): Array<KeyValue<K, V>> | null;
    // (undocumented)
    transform<K extends number, V>(input: Record<K, V> | null | undefined,
    compareFn?: (a: KeyValue<string, V>, b: KeyValue<string, V>) => number):
    Array<KeyValue<string, V>> | null;
    // (undocumented)
    transform<K extends string, V>(input: Record<K, V> | ReadonlyMap<K, V> | null |
    undefined, compareFn?: (a: KeyValue<K, V>, b: KeyValue<K, V>) => number):
    Array<KeyValue<K, V>> | null;
    // (undocumented)
    static efac: i0.eeFactoryDeclaration<KeyValuePipe, never>;
    // (undocumented)
    static epipe: i0.eePipeDeclaration<KeyValuePipe, "keyvalue">;
}

// @public
class Location_2 implements OnDestroy {
    constructor(platformStrategy: LocationStrategy, platformLocation:
    PlatformLocation);
}

```

```

    back(): void;
    forward(): void;
    getState(): unknown;
    go(path: string, query?: string, state?: any): void;
    historyGo(relativePosition?: number): void;
    isCurrentPathEqualTo(path: string, query?: string): boolean;
    static joinWithSlash: (start: string, end: string) => string;
    // (undocumented)
    ngOnDestroy(): void;
    normalize(url: string): string;
    static normalizeQueryParams: (params: string) => string;
    onUrlChange(fn: (url: string, state: unknown) => void): VoidFunction;
    path(includeHash?: boolean): string;
    prepareExternalUrl(url: string): string;
    replaceState(path: string, query?: string, state?: any): void;
    static stripTrailingSlash: (url: string) => string;
    subscribe(onNext: (value: PopStateEvent_2) => void, onThrow?: ((exception: any)
=> void) | null, onReturn?: (() => void) | null): SubscriptionLike;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<Location_2, never>;
    // (undocumented)
    static eprov: i0.ɵɵInjectableDeclaration<Location_2>;
}
export { Location_2 as Location }

// @public
export const LOCATION_INITIALIZED: InjectionToken<Promise<any>>;

// @public
export interface LocationChangeEvent {
    // (undocumented)
    state: any;
    // (undocumented)
    type: string;
}

// @public (undocumented)
export interface LocationChangeListener {
    // (undocumented)
    (event: LocationChangeEvent): any;
}

// @public
export abstract class LocationStrategy {
    // (undocumented)
    abstract back(): void;
    // (undocumented)
    abstract forward(): void;
    // (undocumented)
    abstract getBaseHref(): string;
    // (undocumented)
    historyGo?(relativePosition: number): void;

```



```

    // (undocumented)
    abstract onPopState(fn: LocationChangeListener): void;
    // (undocumented)
    abstract path(includeHash?: boolean): string;
    // (undocumented)
    abstract prepareExternalUrl(internal: string): string;
    // (undocumented)
    abstract pushState(state: any, title: string, url: string, queryParams: string):
void;
    // (undocumented)
    abstract replaceState(state: any, title: string, url: string, queryParams:
string): void;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<LocationStrategy, never>;
    // (undocumented)
    static eprov: i0.ɵɵInjectableDeclaration<LocationStrategy>;
}

// @public
export class LowerCasePipe implements PipeTransform {
    // (undocumented)
    transform(value: string): string;
    // (undocumented)
    transform(value: null | undefined): null;
    // (undocumented)
    transform(value: string | null | undefined): string | null;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<LowerCasePipe, never>;
    // (undocumented)
    static epipe: i0.ɵɵPipeDeclaration<LowerCasePipe, "lowercase">;
}

// @public
export class NgClass implements DoCheck {
    constructor(_iterableDiffers: IterableDiffers, _keyValueDiffers:
KeyValueDiffers, _ngEl: ElementRef, _renderer: Renderer2);
    // (undocumented)
    set klass(value: string);
    // (undocumented)
    set ngClass(value: string | string[] | Set<string> | {
        [klass: string]: any;
    });
    // (undocumented)
    ngDoCheck(): void;
    // (undocumented)
    static edir: i0.ɵɵDirectiveDeclaration<NgClass, "[ngClass]", never, { "klass":
"class"; "ngClass": "ngClass"; }, {}, never>;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<NgClass, never>;
}

// @public

```

```

export class NgComponentOutlet implements OnChanges, OnDestroy {
    constructor(_viewContainerRef: ViewContainerRef);
    // (undocumented)
    ngComponentOutlet: Type<any>;
    // (undocumented)
    ngComponentOutletContent?: any[][];
    // (undocumented)
    ngComponentOutletInjector?: Injector;
    // (undocumented)
    ngComponentOutletNgModule?: Type<any>;
    // @deprecated (undocumented)
    ngComponentOutletNgModuleFactory?: NgModuleFactory<any>;
    // (undocumented)
    ngOnChanges(changes: SimpleChanges): void;
    // (undocumented)
    ngOnDestroy(): void;
    // (undocumented)
    static edir: i0.ɵɵDirectiveDeclaration<NgComponentOutlet, "[ngComponentOutlet]",
never, { "ngComponentOutlet": "ngComponentOutlet"; "ngComponentOutletInjector":
"ngComponentOutletInjector"; "ngComponentOutletContent": "ngComponentOutletContent";
"ngComponentOutletNgModule": "ngComponentOutletNgModule";
"ngComponentOutletNgModuleFactory": "ngComponentOutletNgModuleFactory"; }, {},
never>;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<NgComponentOutlet, never>;
}

// @public
export class NgForOf<T, U extends NgIterable<T> = NgIterable<T>> implements DoCheck
{
    constructor(_viewContainer: ViewContainerRef, _template:
TemplateRef<NgForOfContext<T, U>>, _differs: IterableDiffers);
    ngDoCheck(): void;
    set ngForOf(ngForOf: U & NgIterable<T> | undefined | null);
    set ngForTemplate(value: TemplateRef<NgForOfContext<T, U>>);
    set ngForTrackBy(fn: TrackByFunction<T>);
    // (undocumented)
    get ngForTrackBy(): TrackByFunction<T>;
    static ngTemplateContextGuard<T, U extends NgIterable<T>>(dir: NgForOf<T, U>,
ctx: any): ctx is NgForOfContext<T, U>;
    // (undocumented)
    static edir: i0.ɵɵDirectiveDeclaration<NgForOf<any, any>, "[ngFor][ngForOf]",
never, { "ngForOf": "ngForOf"; "ngForTrackBy": "ngForTrackBy"; "ngForTemplate":
"ngForTemplate"; }, {}, never>;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<NgForOf<any, any>, never>;
}

// @public (undocumented)
export class NgForOfContext<T, U extends NgIterable<T> = NgIterable<T>> {
    // (undocumented)
    $implicit: T;

```

```

    constructor($implicit: T, ngForOf: U, index: number, count: number);
    // (undocumented)
    count: number;
    // (undocumented)
    get even(): boolean;
    // (undocumented)
    get first(): boolean;
    // (undocumented)
    index: number;
    // (undocumented)
    get last(): boolean;
    // (undocumented)
    ngForOf: U;
    // (undocumented)
    get odd(): boolean;
}

// @public
export class NgIf<T = unknown> {
    constructor(_viewContainer: ViewContainerRef, templateRef:
TemplateRef<NgIfContext<T>>);
    set ngIf(condition: T);
    set ngIfElse(templateRef: TemplateRef<NgIfContext<T>> | null);
    set ngIfThen(templateRef: TemplateRef<NgIfContext<T>> | null);
    static ngTemplateContextGuard<T>(dir: NgIf<T>, ctx: any): ctx is
NgIfContext<Exclude<T, false | 0 | '' | null | undefined>>;
    static ngTemplateGuard_ngIf: 'binding';
    // (undocumented)
    static ɵdir: i0.ɵɵDirectiveDeclaration<NgIf<any>, "[ngIf]", never, { "ngIf":
"ngIf"; "ngIfThen": "ngIfThen"; "ngIfElse": "ngIfElse"; }, {}, never>;
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<NgIf<any>, never>;
}

// @public (undocumented)
export class NgIfContext<T = unknown> {
    // (undocumented)
    $implicit: T;
    // (undocumented)
    ngIf: T;
}

// @public
export class NgLocaleLocalization extends NgLocalization {
    constructor(locale: string);
    // (undocumented)
    getPluralCategory(value: any, locale?: string): string;
    // (undocumented)
    protected locale: string;
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<NgLocaleLocalization, never>;
    // (undocumented)

```

```

    static eprov: i0.ɵɵInjectableDeclaration<NgLocaleLocalization>;
}

// @public (undocumented)
export abstract class NgLocalization {
    // (undocumented)
    abstract getPluralCategory(value: any, locale?: string): string;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<NgLocalization, never>;
    // (undocumented)
    static eprov: i0.ɵɵInjectableDeclaration<NgLocalization>;
}

// @public
export class NgPlural {
    constructor(_localization: NgLocalization);
    // (undocumented)
    addCase(value: string, switchView: SwitchView): void;
    // (undocumented)
    set ngPlural(value: number);
    // (undocumented)
    static edir: i0.ɵɵDirectiveDeclaration<NgPlural, "[ngPlural]", never, {
"ngPlural": "ngPlural"; }, {}, never>;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<NgPlural, never>;
}

// @public
export class NgPluralCase {
    constructor(value: string, template: TemplateRef<Object>, viewContainer:
ViewContainerRef, ngPlural: NgPlural);
    // (undocumented)
    value: string;
    // (undocumented)
    static edir: i0.ɵɵDirectiveDeclaration<NgPluralCase, "[ngPluralCase]", never,
{}, {}, never>;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<NgPluralCase, [{ attribute: "ngPluralCase";
}, null, null, { host: true; }]>;
}

// @public
export class NgStyle implements DoCheck {
    constructor(_ngEl: ElementRef, _differs: KeyValueDiffers, _renderer: Renderer2);
    // (undocumented)
    ngDoCheck(): void;
    // (undocumented)
    set ngStyle(values: {
        [klass: string]: any;
    } | null);
    // (undocumented)
    static edir: i0.ɵɵDirectiveDeclaration<NgStyle, "[ngStyle]", never, { "ngStyle":

```

```

"ngStyle"; }, {}, never>;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<NgStyle, never>;
}

// @public
export class NgSwitch {
    // (undocumented)
    set ngSwitch(newValue: any);
    // (undocumented)
    static edir: i0.ɵɵDirectiveDeclaration<NgSwitch, "[ngSwitch]", never, {
"ngSwitch": "ngSwitch"; }, {}, never>;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<NgSwitch, never>;
}

// @public
export class NgSwitchCase implements DoCheck {
    constructor(viewContainer: ViewContainerRef, templateRef: TemplateRef<Object>,
ngSwitch: NgSwitch);
    ngDoCheck(): void;
    ngSwitchCase: any;
    // (undocumented)
    static edir: i0.ɵɵDirectiveDeclaration<NgSwitchCase, "[ngSwitchCase]", never, {
"ngSwitchCase": "ngSwitchCase"; }, {}, never>;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<NgSwitchCase, [null, null, { optional:
true; host: true; }]>;
}

// @public
export class NgSwitchDefault {
    constructor(viewContainer: ViewContainerRef, templateRef: TemplateRef<Object>,
ngSwitch: NgSwitch);
    // (undocumented)
    static edir: i0.ɵɵDirectiveDeclaration<NgSwitchDefault, "[ngSwitchDefault]",
never, {}, {}, never>;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<NgSwitchDefault, [null, null, { optional:
true; host: true; }]>;
}

// @public
export class NgTemplateOutlet implements OnChanges {
    constructor(_viewContainerRef: ViewContainerRef);
    // (undocumented)
    ngOnChanges(changes: SimpleChanges): void;
    ngTemplateOutlet: TemplateRef<any> | null;
    ngTemplateOutletContext: Object | null;
    ngTemplateOutletInjector: Injector | null;
    // (undocumented)
    static edir: i0.ɵɵDirectiveDeclaration<NgTemplateOutlet, "[ngTemplateOutlet]",

```

```

never, { "ngTemplateOutletContext": "ngTemplateOutletContext"; "ngTemplateOutlet":
"ngTemplateOutlet"; "ngTemplateOutletInjector": "ngTemplateOutletInjector"; }, {},
never>;
    // (undocumented)
    static efac: i0.eefactoryDeclaration<NgTemplateOutlet, never>;
}

// @public
export enum NumberFormatStyle {
    // (undocumented)
    Currency = 2,
    // (undocumented)
    Decimal = 0,
    // (undocumented)
    Percent = 1,
    // (undocumented)
    Scientific = 3
}

// @public
export enum NumberSymbol {
    CurrencyDecimal = 12,
    CurrencyGroup = 13,
    Decimal = 0,
    Exponential = 6,
    Group = 1,
    Infinity = 9,
    List = 2,
    MinusSign = 5,
    NaN = 10,
    PercentSign = 3,
    PerMille = 8,
    PlusSign = 4,
    SuperscriptingExponent = 7,
    TimeSeparator = 11
}

// @public
export class PathLocationStrategy extends LocationStrategy implements OnDestroy {
    constructor(_platformLocation: PlatformLocation, href?: string);
    // (undocumented)
    back(): void;
    // (undocumented)
    forward(): void;
    // (undocumented)
    getBaseHref(): string;
    // (undocumented)
    historyGo(relativePosition?: number): void;
    // (undocumented)
    ngOnDestroy(): void;
    // (undocumented)
    onPopState(fn: LocationChangeListener): void;
}

```

```

    // (undocumented)
    path(includeHash?: boolean): string;
    // (undocumented)
    prepareExternalUrl(internal: string): string;
    // (undocumented)
    pushState(state: any, title: string, url: string, queryParams: string): void;
    // (undocumented)
    replaceState(state: any, title: string, url: string, queryParams: string): void;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<PathLocationStrategy, [null, { optional:
true; }]>;
    // (undocumented)
    static eprov: i0.ɵɵInjectableDeclaration<PathLocationStrategy>;
}

// @public
export class PercentPipe implements PipeTransform {
    constructor(_locale: string);
    // (undocumented)
    transform(value: number | string, digitsInfo?: string, locale?: string): string
| null;
    // (undocumented)
    transform(value: null | undefined, digitsInfo?: string, locale?: string): null;
    // (undocumented)
    transform(value: number | string | null | undefined, digitsInfo?: string,
locale?: string): string | null;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<PercentPipe, never>;
    // (undocumented)
    static epipe: i0.ɵɵPipeDeclaration<PercentPipe, "percent">;
}

// @public
export abstract class PlatformLocation {
    // (undocumented)
    abstract back(): void;
    // (undocumented)
    abstract forward(): void;
    // (undocumented)
    abstract getBaseHrefFromDOM(): string;
    // (undocumented)
    abstract getState(): unknown;
    // (undocumented)
    abstract get hash(): string;
    // (undocumented)
    historyGo?(relativePosition: number): void;
    // (undocumented)
    abstract get hostname(): string;
    // (undocumented)
    abstract get href(): string;
    abstract onHashChange(fn: LocationChangeListener): VoidFunction;
    abstract onPopState(fn: LocationChangeListener): VoidFunction;

```

```

    // (undocumented)
    abstract get pathname(): string;
    // (undocumented)
    abstract get port(): string;
    // (undocumented)
    abstract get protocol(): string;
    // (undocumented)
    abstract pushState(state: any, title: string, url: string): void;
    // (undocumented)
    abstract replaceState(state: any, title: string, url: string): void;
    // (undocumented)
    abstract get search(): string;
    // (undocumented)
    static efac: i0.ɵɵFactoryDeclaration<PlatformLocation, never>;
    // (undocumented)
    static eprov: i0.ɵɵInjectableDeclaration<PlatformLocation>;
}

// @public
export enum Plural {
    // (undocumented)
    Few = 3,
    // (undocumented)
    Many = 4,
    // (undocumented)
    One = 1,
    // (undocumented)
    Other = 5,
    // (undocumented)
    Two = 2,
    // (undocumented)
    Zero = 0
}

// @public (undocumented)
interface PopStateEvent_2 {
    // (undocumented)
    pop?: boolean;
    // (undocumented)
    state?: any;
    // (undocumented)
    type?: string;
    // (undocumented)
    url?: string;
}
export { PopStateEvent_2 as PopStateEvent }

// @public
export function registerLocaleData(data: any, localeId?: string | any, extraData?:
any): void;

// @public

```



```

export class SlicePipe implements PipeTransform {
    // (undocumented)
    transform<T>(value: ReadonlyArray<T>, start: number, end?: number): Array<T>;
    // (undocumented)
    transform(value: null | undefined, start: number, end?: number): null;
    // (undocumented)
    transform<T>(value: ReadonlyArray<T> | null | undefined, start: number, end?:
number): Array<T> | null;
    // (undocumented)
    transform(value: string, start: number, end?: number): string;
    // (undocumented)
    transform(value: string | null | undefined, start: number, end?: number): string
| null;
    // (undocumented)
    static efac: i0.eeFactoryDeclaration<SlicePipe, never>;
    // (undocumented)
    static epipe: i0.eePipeDeclaration<SlicePipe, "slice">;
}

// @public
export type Time = {
    hours: number;
    minutes: number;
};

// @public
export class TitleCasePipe implements PipeTransform {
    // (undocumented)
    transform(value: string): string;
    // (undocumented)
    transform(value: null | undefined): null;
    // (undocumented)
    transform(value: string | null | undefined): string | null;
    // (undocumented)
    static efac: i0.eeFactoryDeclaration<TitleCasePipe, never>;
    // (undocumented)
    static epipe: i0.eePipeDeclaration<TitleCasePipe, "titlecase">;
}

// @public
export enum TranslationWidth {
    Abbreviated = 1,
    Narrow = 0,
    Short = 3,
    Wide = 2
}

// @public
export class UpperCasePipe implements PipeTransform {
    // (undocumented)
    transform(value: string): string;
    // (undocumented)

```

```

    transform(value: null | undefined): null;
    // (undocumented)
    transform(value: string | null | undefined): string | null;
    // (undocumented)
    static efac: i0.eeFactoryDeclaration<UpperCasePipe, never>;
    // (undocumented)
    static epipe: i0.eePipeDeclaration<UpperCasePipe, "uppercase">;
}

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

// @public
export abstract class ViewportScroller {
    abstract getScrollPosition(): [number, number];
    abstract scrollToAnchor(anchor: string): void;
    abstract scrollToPosition(position: [number, number]): void;
    abstract setHistoryScrollRestoration(scrollRestoration: 'auto' | 'manual'):
void;
    abstract setOffset(offset: [number, number] | (() => [number, number])): void;
    // (undocumented)
    static eprov: unknown;
}

// @public
export enum WeekDay {
    // (undocumented)
    Friday = 5,
    // (undocumented)
    Monday = 1,
    // (undocumented)
    Saturday = 6,
    // (undocumented)
    Sunday = 0,
    // (undocumented)
    Thursday = 4,
    // (undocumented)
    Tuesday = 2,
    // (undocumented)
    Wednesday = 3
}

// @public
export abstract class XhrFactory {
    // (undocumented)
    abstract build(): XMLHttpRequest;
}

// (No @packageDocumentation comment for this package)

```