API Report File for “@angular/animations”

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

```
// @public
export function animate(timings: string | number, styles?: AnimationStyleMetadata | Animatio

// @public
export function animateChild(options?: AnimateChildOptions | null): AnimationAnimateChildMet

// @public
export interface AnimateChildOptions extends AnimationOptions {
    // (undocumented)
    duration?: number | string;
}

// @public
export type AnimateTimings = {
    duration: number;
    delay: number;
    easing: string | null;
};

// @public
export function animation(steps: AnimationMetadata | AnimationMetadata[], options?: Animatio

// @public
export interface AnimationAnimateChildMetadata extends AnimationMetadata {
    options: AnimationOptions | null;
}

// @public
export interface AnimationAnimateMetadata extends AnimationMetadata {
    styles: AnimationStyleMetadata | AnimationKeyframesSequenceMetadata | null;
    timings: string | number | AnimateTimings;
}

// @public
export interface AnimationAnimateRefMetadata extends AnimationMetadata {
    animation: AnimationReferenceMetadata;
    options: AnimationOptions | null;
}

// @public
export abstract class AnimationBuilder {
    abstract build(animation: AnimationMetadata | AnimationMetadata[]): AnimationFactory;
}

// @public
interface AnimationEvent_2 {
    disabled: boolean;
    element: any;
    fromState: string;
    phaseName: string;
    toState: string;
    totalTime: number;
    triggerName: string;
}
export { AnimationEvent_2 as AnimationEvent }

// @public
export abstract class AnimationFactory {
    abstract create(element: any, options?: AnimationOptions): AnimationPlayer;
}

// @public
export interface AnimationGroupMetadata extends AnimationMetadata {
    options: AnimationOptions | null;
    steps: AnimationMetadata[];
}

// @public
export interface AnimationKeyframesSequenceMetadata extends AnimationMetadata {
    steps: AnimationStyleMetadata[];
}

// @public
export interface AnimationMetadata {
    // (undocumented)
    type: AnimationMetadataType;
}

// @public
export const enum AnimationMetadataType {
    Animate = 4,
    AnimateChild = 9,
    AnimateRef = 10,
    Group = 3,
    Keyframes = 5,
    Query = 11,
    Reference = 8,
    Sequence = 2,
    Stagger = 12,
    State = 0,
    Style = 6,
    Transition = 1,
    Trigger = 7
}

// @public
export interface AnimationOptions {
    delay?: number | string;
    params?: {
        [name: string]: any;
    };
}

// @public
export interface AnimationPlayer {
    beforeDestroy?: () => any;
    destroy(): void;
    finish(): void;
    getPosition(): number;
    hasStarted(): boolean;
    init(): void;
    onDestroy(fn: () => void): void;
    onDone(fn: () => void): void;
    onStart(fn: () => void): void;
    parentPlayer: AnimationPlayer | null;
    pause(): void;
    play(): void;
    reset(): void;
    restart(): void;
    setPosition(position: any /** TODO #9100 */): void;
    readonly totalTime: number;
}

// @public
export interface AnimationQueryMetadata extends AnimationMetadata {
    animation: AnimationMetadata | AnimationMetadata[];
    options: AnimationQueryOptions | null;
    selector: string;
}

// @public
export interface AnimationQueryOptions extends AnimationOptions {
    limit?: number;
    optional?: boolean;
}

// @public
export interface AnimationReferenceMetadata extends AnimationMetadata {
    animation: AnimationMetadata | AnimationMetadata[];
    options: AnimationOptions | null;
}

// @public
export interface AnimationSequenceMetadata extends AnimationMetadata {
    options: AnimationOptions | null;
    steps: AnimationMetadata[];
}

// @public
export interface AnimationStaggerMetadata extends AnimationMetadata {
    animation: AnimationMetadata | AnimationMetadata[];
    timings: string | number;
}

// @public
export interface AnimationStateMetadata extends AnimationMetadata {
    name: string;
    options?: {
        params: {
            [name: string]: any;
        };
    };
    styles: AnimationStyleMetadata;
}

// @public
export interface AnimationStyleMetadata extends AnimationMetadata {
    offset: number | null;
    styles: '*' | {
        [key: string]: string | number;
    } | Array<{
        [key: string]: string | number;
    } | '*'>;
}

// @public
export interface AnimationTransitionMetadata extends AnimationMetadata {
    animation: AnimationMetadata | AnimationMetadata[];
    expr: string | ((fromState: string, toState: string, element?: any, params?: {
        [key: string]: any;
    }) => boolean);
    options: AnimationOptions | null;
}

// @public
export interface AnimationTriggerMetadata extends AnimationMetadata {
    definitions: AnimationMetadata[];
    name: string;
    options: {
        params?: {
            [name: string]: any;
        };
    } | null;
}

// @public
export const AUTO_STYLE = "*";

// @public
export function group(steps: AnimationMetadata[], options?: AnimationOptions | null): Animat

// @public
export function keyframes(steps: AnimationStyleMetadata[]): AnimationKeyframesSequenceMetada

// @public
export class NoopAnimationPlayer implements AnimationPlayer {
    constructor(duration?: number, delay?: number);
    // (undocumented)
    destroy(): void;
    // (undocumented)
    finish(): void;
    // (undocumented)
    getPosition(): number;
    // (undocumented)
    hasStarted(): boolean;
    // (undocumented)
    init(): void;
    // (undocumented)
    onDestroy(fn: () => void): void;
    // (undocumented)
    onDone(fn: () => void): void;
    // (undocumented)
    onStart(fn: () => void): void;
    // (undocumented)
    parentPlayer: AnimationPlayer | null;
    // (undocumented)
    pause(): void;
    // (undocumented)
    play(): void;
    // (undocumented)
    reset(): void;
    // (undocumented)
    restart(): void;
    // (undocumented)
    setPosition(position: number): void;
    // (undocumented)
    readonly totalTime: number;
}

// @public
export function query(selector: string, animation: AnimationMetadata | AnimationMetadata[],

// @public
export function sequence(steps: AnimationMetadata[], options?: AnimationOptions | null): Ani

// @public
export function stagger(timings: string | number, animation: AnimationMetadata | AnimationMe

// @public
export function state(name: string, styles: AnimationStyleMetadata, options?: {
    params: {
        [name: string]: any;
    };
}): AnimationStateMetadata;

// @public
export function style(tokens: '*' | {
    [key: string]: string | number;
} | Array<'*' | {
    [key: string]: string | number;
}>): AnimationStyleMetadata;

// @public
export function transition(stateChangeExpr: string | ((fromState: string, toState: string, e
    [key: string]: any;
}) => boolean), steps: AnimationMetadata | AnimationMetadata[], options?: AnimationOptions

// @public
export function trigger(name: string, definitions: AnimationMetadata[]): AnimationTriggerMet

// @public
```

```
export function useAnimation(animation: AnimationReferenceMetadata, options?: AnimationOptic

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