**Angular 8 - Animations**

[Next](https://www.tutorialspoint.com/angular8/angular8_forms.htm)

Animation gives the web application a refreshing look and rich user interaction. In HTML, animation is basically the transformation of HTML element from one CSS style to another over a specific period of time. For example, an image element can be enlarged by changing its width and height.

If the width and height of the image is changed from initial value to final value in steps over a period of time, say 10 seconds, then we get an animation effect. So, the scope of the animation depends on the feature / property provided by the CSS to style a HTML element.

Angular provides a separate module **BrowserAnimationModule** to do the animation. **BrowserAnimationModule** provides an easy and clear approach to do animation.

Configuring animation module

Let us learn how to configure animation module in this chapter.

Follow below mentioned steps to configure animation module, **BrowserAnimationModule** in an application.

Import **BrowserAnimationModule** in AppModule.

import { BrowserAnimationsModule } from '@angular/platform-browser/animations';

@NgModule({

imports: [

BrowserModule,

BrowserAnimationsModule

],

declarations: [ ],

bootstrap: [ ]

})

export class AppModule { }

Import animation function in the relevant components.

import { state, style, transition, animate, trigger } from '@angular/animations'

Add **animations** metadata property in the relevant component.

@Component({

animations: [

// animation functionality goes here

]

})

export class MyAnimationComponent

Concepts

In angular, we need to understand the five core concept and its relationship to do animation.

State

State refers the specific state of the component. A component can have multiple defined state. The state is created using state() method. state() method has two arguments.

* **name** − Unique name of the state.
* **style** − Style of the state defined using style() method.

animations: [

...

state('start', style( { width: 200px; } ))

...

]

Here, **start** is the name of the state.

Style

**Style** refers the CSS style applied in a particular state. style() method is used to style the particular state of a component. It uses the CSS property and can have multiple items.

animations: [

...

state('start', style( { width: 200px; opacity: 1 } ))

...

]

Here, **start** state defines two CSS property, **width** with value 200px and opacity with value 1.

Transition

**Transition** refers the transition from one state to another. Animation can have multiple transition. Each transition is defined using transition() function. transition() takes two argument.

* Specifies the direction between two transition state. For example, **start => end**refers that the initial state is **start** and the final state is **end**. Actually, it is an expression with rich functionality.
* Specifies the animation details using **animate()** function.

animations: [

...

transition('start => end', [

animate('1s')

])

...

]

Here, **transition()** function defines the transition from start state to end state with animation defined in **animate()** method.

Animation

Animation defines the way the transition from one state to another take place. **animation()** function is used to set the animation details. **animate()** takes a single argument in the form of below expression −

duration delay easing

* **duration** − refers the duration of the transition. It is expressed as 1s, 100ms, etc.,
* **delay** − refers the delay time to start the transition. It is expressed similar to *duration*
* **easing** − refers how do to accelerates / decelerates the transition in the given time duration.

Trigger

Every animation needs a trigger to start the animation. trigger() method is used to set all the animation information such as state, style, transition and animation in one place and give it a unique name. The unique name is used further to trigger the animation.

animations: [

trigger('enlarge', [

state('start', style({

height: '200px',

})),

state('end', style({

height: '500px',

})),

transition('start => end', [

animate('1s')

]),

transition('end => start', [

animate('0.5s')

]) ]),

]

Here, **enlarge** is the unique name given to the particular animation. It has two state and related styles. It has two transition one from start to end and another from end to start. End to start state do the reverse of the animation.

**Trigger** can be attached to an element as specified below −

<div [@triggerName]="expression">...</div>;

For example,

<img [@enlarge]="isEnlarge ? 'end' : 'start'">...</img>;

Here,

* **@enlarge** − trigger is set to image tag and attrached to an expression.
* If **isEnlarge** value is changed to true, then **end** state will be set and it triggers **start =>** end transition.
* If **isEnlarge** value is changed to false, then **start** state will be set and it triggers **end => start transition**.

Simple Animation Example

Let us write a new angular application to better understand the animation concept by enlarging an image with animation effect.

Open command prompt and create new angular application.

cd /go/to/workspace

ng new animation-app

cd animation-app

Configure **BrowserAnimationModule** in the **AppModule** (src/app/app.module.ts)

import { BrowserModule } from '@angular/platform-browser';

import { NgModule } from '@angular/core'

import { BrowserAnimationsModule } from '@angular/platform-browser/animations';

import { AppComponent } from './app.component'; @NgModule({

declarations: [

AppComponent

],

imports: [

BrowserModule,

BrowserAnimationsModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

Open **AppComponent (src/app/app.component.ts)** and import necessary animation functions.

import { state, style, transition, animate, trigger } from '@angular/animations';

Add animation functionality, which will animate the image during the enlarging / shrinking of the image.

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css'],

animations: [

trigger('enlarge', [

state('start', style({

height: '150px'

})),

state('end', style({

height: '250px'

})),

transition('start => end', [

animate('1s 2s')

]),

transition('end => start', [

animate('1s 2s')

])

])

]

})

Open **AppComponent** template, **src/app/app.component.html** and remove sample code. Then, include a header with application title, image and a button to enlarge / shrink the image.

<h1>{{ title }}</h1>

<img src="assets/puppy.jpeg" style="height: 200px" /> <br />

<button>{{ this.buttonText }}</button>

Write a function to change the animation expression.

export class AppComponent {

title = 'Animation Application';

isEnlarge: boolean = false;

buttonText: string = "Enlarge";

triggerAnimation() {

this.isEnlarge = !this.isEnlarge;

if(this.isEnlarge)

this.buttonText = "Shrink";

else

this.buttonText = "Enlarge";

}

}

Attach the animation in the image tag. Also, attach the click event for the button.

<h1>{{ title }}</h1>

<img [@enlarge]="isEnlarge ? 'end' : 'start'" src="assets/puppy.jpeg" style="height: 200px" />

<br />

<button (click)='triggerAnimation()'>{{ this.buttonText }}</button>

The complete **AppComponent** code is as follows −

import { Component } from '@angular/core';

import { state, style, transition, animate, trigger } from '@angular/animations';

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css'],

animations: [

trigger('enlarge', [

state('start', style({

height: '150px'

})),

state('end', style({

height: '250px'

})),

transition('start => end', [

animate('1s 2s')

]),

transition('end => start', [

animate('1s 2s')

])

])

]

})

export class AppComponent {

title = 'Animation Application';

isEnlarge: boolean = false;

buttonText: string = "Enlarge";

triggerAnimation() {

this.isEnlarge = !this.isEnlarge;

if(this.isEnlarge)

this.buttonText = "Shrink";

else

this.buttonText = "Enlarge";

}

}

The complete AppComponent template code is as follows −

<h1>{{ title }}</h1>

<img [@enlarge]="isEnlarge ? 'end' : 'start'" src="assets/puppy.jpeg" style="height: 200px" />

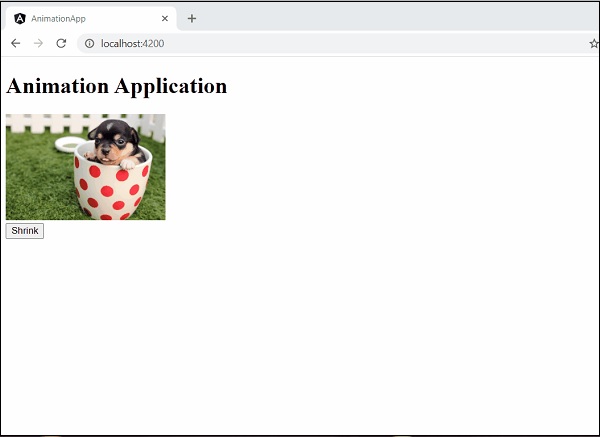
<br />

<button (click)='triggerAnimation()'>{{ this.buttonText }}</button>

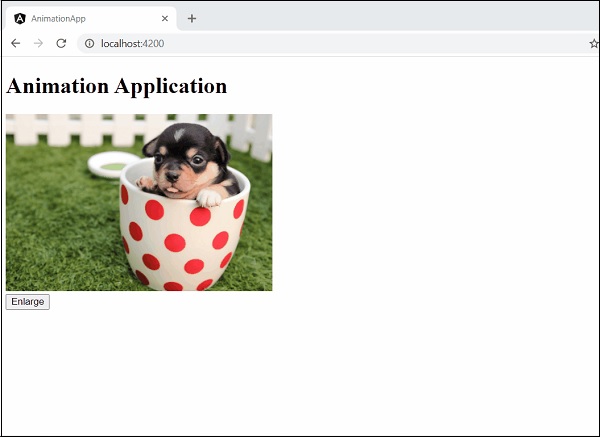
Run the application using below command −

ng serve

Click the enlarge button, it will enlarge the image with animation. The result will be as shown below −



Click the button again to shrink it. The result will be as shown below −



Print Page