Angular Animations

Angular provides a library for configuring the CSS animations dynamically. Angular can implement CSS 2D and 3D effects by using the following:

- CSS “transition”
- CSS “transform”
- CSS “animate”

Angular have animation library **“BrowserAnimationModule”** for dynamically controlling CSS animations. It is defined in **“@angular/animations”.** It provides the following animation methods.

Animation Method	Description
trigger()	It is used to configure the animation effects in a component.
state()	It is used to configure animation state. You have to define 2 animation states a)Initial State

	b)Final State Initial state specifies the effects for element before the transformation. You can represent by using “[void=>*]”. Final state specifies the effects for element after transformation. You can represent by using “[*=>void]”.
style()	It is used to defines CSS effects to any element in component. You can use all CSS attributes.
transition()	It configures the state and the time for animation.
animate()	It configures the animation duration.

- All animation functions are derived from “@angular/animations”

- Angular animations module is added into your project with “angular material”.
- Animations for any component are configured in the component meta data by using “animations[]”.

Syntax:

```
@Component({
      Selector: ‘ ‘,
      templateUrl: ‘ ‘,
      styleUrls: [ ],
      animations: [ ]
      })
```

Animations collection uses animation functions:

Syntax:

```
animations:[
  trigger(‘EffectName’, [
      state(‘initial’, style({ attribute:value })),
      state(‘final’, style({ attribute:value })),
      transition(‘initial=>final/void=>*’,
animate(time)),
```

transition('final=>initial/*=>void, animate(time))
]) // end of trigger
]

You have to apply your trigger any HTML element in component.
<div [@TriggerName] >

1. Add angular Material Library for "Animations" module
2. Add a new component
3. Animationsdemo.component.ts

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

@Component({
  selector: 'app-animationsdemo',
```

```
  templateUrl: './animationsdemo.component.html',
  styleUrls: ['./animationsdemo.component.css'],
  animations: [
     trigger('ZoomEffect', [
       state('initial', style({
         width: '200px',
         height: '200px',
         transform: 'rotate(0deg)'
       })),
       state('final', style({
         width: '400px',
         height: '400px',
         transform: 'rotate(360deg)'
       })),
       transition('initial=>final', animate('3000ms')),
       transition('final=>initial', animate('3000ms'))
     ])
  ]
```

```
})
export class AnimationsdemoComponent
 implements OnInit {

  public animationState = 'initial';
  public zoomText = 'Zoom In';
  constructor() { }

  ngOnInit(): void {
  }
  public ZoomClick() {
    this.animationState = (this.anim
ationState == 'initial')?'final':'in
itial';
    this.zoomText = (this.zoomText==
 'Zoom In')? 'Zoom Out': 'Zoom In';
  }
}
```

4. Animationsdemo.component.html

```
<div class="container">
```

```
    <div class="form-group">
        <button (click)="ZoomClick()
">{{zoomText}}</button>
    </div>
    <div class="form-group">
        <img [@ZoomEffect]="animatio
nState" src="assets/shirt.jpg">
    </div>
</div>
```

Angular Testing

- Angular supports frameworks like MVC and MVVM.
- MVC and MVVM enable unit testing.
- Unit testing include testing every function that your write for component, service, pipe etc.
- Testing verifies that the expected values and returned value are same. It reports bugs if the returned value is not same as expected.
- Angular is integrated with Jasmin & Karma

Ex:

1. Create a new Project Repository
2. Open in VS Code
3. Install Jasmine Framework

   > npm install jasmine-core

4. Add a new HTML page “index.html”

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="node_modules/jasmine-core/lib/jasmine-core/jasmine.css">
    <script src="node_modules/jasmine-core/lib/jasmine-core/jasmine.js"></script>
    <script src="node_modules/jasmine-core/lib/jasmine-core/jasmine-html.js"></script>
    <script src="node_modules/jasmine-core/lib/jasmine-core/boot.js"></script>
  </head>
</html>
```

- **Testing Every component, service or pipe function include 3 stages**
  - Arrange : describe()
  - Act : it()
  - Assert : expect().toBe()

  expect() is what developer written.

  toBe() is what client requires.

Ex:

1. Add a new folder “Components”
2. Add following files

   Math.component.js

```
function addition(a, b)
{
   return a + b;
}
function hello(str) {
   return str;
}
```

   Math.component.spec.js

```
describe("MathComponentTest", function(){
```

```
    it("Addition Test", function(){
        expect(addition(20,20)).toBe(40);
    })
    it("Hello Test", function(){
        expect(hello("John")).toBe("Johns");
    })
})
```

Index.html

```
<!DOCTYPE html>
<html>
    <head>
        <link rel="stylesheet"
href="node_modules/jasmine-
core/lib/jasmine-core/jasmine.css">
        <script src="node_modules/jasmine-
core/lib/jasmine-core/jasmine.js"></script>
        <script src="node_modules/jasmine-
core/lib/jasmine-core/jasmine-
html.js"></script>
        <script src="node_modules/jasmine-
core/lib/jasmine-core/boot.js"></script>
        <script
src="Components/math.component.js"></scri
pt>
```

```
        <script
    src="Components/math.component.spec.js">
    </script>
       </head>
    </html>
```

Angular Project test:

> ng test -- project = Amazon

- Angular Material – Date Picker – new Angular 10.
- Strict Mode is new for
  ng new –strict
- TypeScript 3.9
- ESM5 [ECMA 5] [ECMA 6]
- More Browsers

ng update @angular/cli @angular/core