

Module 11: Testing and Application Deployment in Angular

Demo Document 1: Unit testing using Jasmin and Karma

edureka!

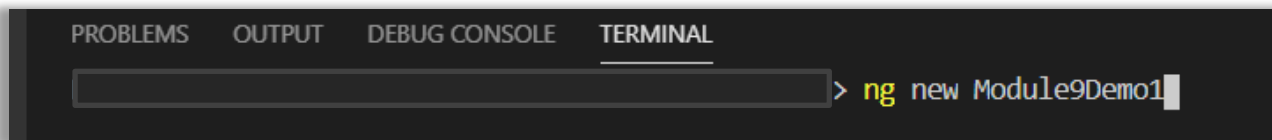
edureka!

© Brain4ce Education Solutions Pvt. Ltd.

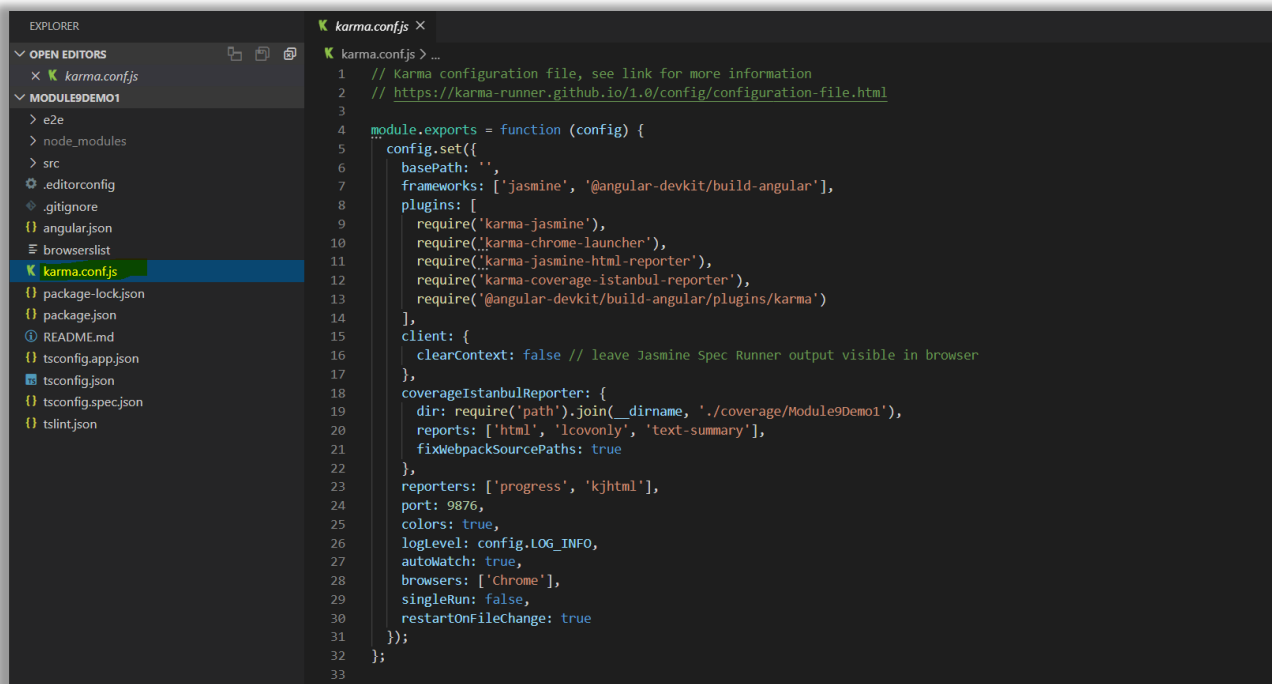
Creating unit testing steps for angular application using karma & jasmine

In this demo, we will see how to create and execute Unit testing of angular app with the help of karma & jasmine.

Step 1: Create new project using command 'ng new Module11Demo1'.



Step 2: When we create new app in angular, angular cli will automatically create karma.config.js file at root location



Step 3- When you create the project all the dependencies get installed. Open package.json file

```

11  },
12  "private": true,
13  "dependencies": {
14    "@angular/animations": "~8.2.4",
15    "@angular/common": "~8.2.4",
16    "@angular/compiler": "~8.2.4",
17    "@angular/core": "~8.2.4",
18    "@angular/forms": "~8.2.4",
19    "@angular/platform-browser": "~8.2.4",
20    "@angular/platform-browser-dynamic": "~8.2.4",
21    "@angular/router": "~8.2.4",
22    "rxjs": "~6.4.0",
23    "tslib": "^1.10.0",
24    "zone.js": "~0.9.1"
25  },
26  "devDependencies": {
27    "@angular-devkit/build-angular": "~0.803.2",
28    "@angular/cli": "~8.3.2",
29    "@angular/compiler-cli": "~8.2.4",
30    "@angular/language-service": "~8.2.4",
31    "@types/node": "~8.9.4",
32    "@types/jasmine": "~3.3.8",
33    "@types/jasminewd2": "~2.0.3",
34    "codifyer": "^5.0.0",
35    "jasmine-core": "~3.4.0",
36    "jasmine-spec-reporter": "~4.2.1",
37    "karma": "~4.1.0",
38    "karma-chrome-launcher": "~2.2.0",
39    "karma-coverage-istanbul-reporter": "~2.0.1",
40    "karma-jasmine": "~2.0.1",
41    "karma-jasmine-html-reporter": "^1.4.0",
42    "protractor": "~5.4.0",
43    "ts-node": "~7.0.0",
44    "tslint": "~5.15.0",
45    "typescript": "~3.5.3"
  }

```

Step 4 – The angular-cli configuration of karma uses the file “test.ts” as the entry point to test the application. Open test.ts file

```

src > TS test.ts > ...
1  // This file is required by karma.conf.js and loads recursively all the .spec and framework files
2
3  import 'zone.js/dist/zone-testing';
4  import { getTestBed } from '@angular/core/testing';
5  import {
6    BrowserDynamicTestingModule,
7    platformBrowserDynamicTesting
8  } from '@angular/platform-browser-dynamic/testing';
9
10 declare const require: any;
11
12 // First, initialize the Angular testing environment.
13 getTestBed().initTestEnvironment(
14   BrowserDynamicTestingModule,
15   platformBrowserDynamicTesting()
16 );
17 // Then we find all the tests.
18 const context = require.context('./', true, /\.spec\.ts$/);
19 // And load the modules.
20 context.keys().map(context);
21

```

Step 5- Create first test, open app.component.html and remove all html code except router-outlet as below

```
<> app.component.html •
src > app > <> app.component.html > ...
1 |
2 | <router-outlet></router-outlet>
```

Step 6- Open inde.html and add bootstrap reference

```
<> app.component.html • <> index.html •
src > <> index.html > html > head > link
1 | <!doctype html>
2 | <html lang="en">
3 | <head>
4 |   <meta charset="utf-8">
5 |   <title>Module9Demo1</title>
6 |   <base href="/">
7 |   <meta name="viewport" content="width=device-width, initial-scale=1">
8 |   <link rel="icon" type="image/x-icon" href="favicon.ico">
9 |   <link href="//cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.3.1/css/bootstrap.min.css" rel="stylesheet" />
10 | </head>
11 | <body>
12 |   <app-root></app-root>
13 | </body>
14 | </html>
15 |
```

Step 7- Ppen app.component.html and add tile and styles

```
<> app.component.html × # app.component.css <> index.html
src > app > <> app.component.html > div.content > div.card.highlight-card.card
1 | <style>
2 |   .content {
3 |     display: flex;
4 |     margin: 32px auto;
5 |     padding: 0 16px;
6 |     max-width: 960px;
7 |     flex-direction: column;
8 |     align-items: center;
9 |   }
10 | </style>
11 |
12 | <div class="content" role="main">
13 |
14 |   <div class="card highlight-card card-small">
15 |     <span>{{ title }} app is running!</span>
16 |   </div>
17 |
18 |
19 |
20 | <router-outlet></router-outlet>
```

Step 8- Open app.component.spec.ts (where we will write test cases). It has already written some test cases as below.

```

src > app > TS app.component.spec.ts > describe('AppComponent') callback > it('should have as title 'Module9Demo1') callback
1  import { TestBed, async } from '@angular/core/testing';
2  import { RouterTestingModule } from '@angular/router/testing';
3  import { AppComponent } from './app.component';
4
5  describe('AppComponent', () => {
6    beforeEach(async(() => {
7      TestBed.configureTestingModule({
8        imports: [
9          RouterTestingModule
10         ],
11         declarations: [
12           AppComponent
13         ],
14       }).compileComponents();
15     }));
16
17     it('should create the app', () => {
18       const fixture = TestBed.createComponent(AppComponent);
19       const app = fixture.debugElement.componentInstance;
20       expect(app).toBeTruthy();
21     });
22
23     it('should have as title 'Module9Demo1'', () => {
24       const fixture = TestBed.createComponent(AppComponent);
25       const app = fixture.debugElement.componentInstance;
26       expect(app.title).toEqual('Module9Demo1');
27     });
28
29     it('should render title', () => {
30       const fixture = TestBed.createComponent(AppComponent);
31       fixture.detectChanges();
32       const compiled = fixture.debugElement.nativeElement;
33       expect(compiled.querySelector('.content span').textContent).toContain('Module9Demo1 app is running!');
34     });
35   });
36

```

Step 9- As we have already written test case for title, run test using command ‘ng test’ as below.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\Work\TapChief\Edureka\Angular8Demo\Module9\Module9Demo1> ng test

```

Chrome will open below window and show how many test cases passed.

Step 10 – Open app.component.ts file and change title of app from ‘Module9Demo1’ to something else as below.

```

src > app > TS app.component.ts > ...
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-root',
5    templateUrl: './app.component.html',
6    styleUrls: ['./app.component.css']
7  })
8  export class AppComponent {
9    title = 'DummyTitle';
10 }
11

```

Step 11 – Save file and run command ng test again.

2 specs will fail as we have modified title.

```

localhost:9876/?id=24200871
Karma v4.1.0 - connected
Chrome 79.0.3945 (Windows 10.0.0) is idle
Jasmine 3.4.0
• x x
3 specs, 2 failures, randomized with seed 93762
Spec List | Failures
AppComponent > should have as title 'Module9Demo1'
Expected 'DummyTitle' to equal 'Module9Demo1'.
Error: Expected 'DummyTitle' to equal 'Module9Demo1'.
    at <Jasmine>
    at UserContext.<anonymous> (http://localhost:9876/_karma_webpack_/src/app/app.component.spec.ts:26:23)
    at ZoneDelegate.invoke (http://localhost:9876/_karma_webpack_/node_modules/zone.js/dist/zone-evergreen.js:359:1)
    at ProxyZoneSpec.push../node_modules/zone.js/dist/zone-testing.js.ProxyZoneSpec.onInvoke (http://localhost:9876/_karma_webpack_/node_modules/zone.js/dist/zone-testing.js:308:1)
AppComponent > should render title
Expected 'DummyTitle app is running!' to contain 'Module9Demo1 app is running!'.
Error: Expected 'DummyTitle app is running!' to contain 'Module9Demo1 app is running!'.
    at <Jasmine>
    at UserContext.<anonymous> (http://localhost:9876/_karma_webpack_/src/app/app.component.spec.ts:33:65)
    at ZoneDelegate.invoke (http://localhost:9876/_karma_webpack_/node_modules/zone.js/dist/zone-evergreen.js:359:1)
    at ProxyZoneSpec.push../node_modules/zone.js/dist/zone-testing.js.ProxyZoneSpec.onInvoke (http://localhost:9876/_karma_webpack_/node_modules/zone.js/dist/zone-testing.js:308:1)

DummyTitle app is running!

```

Similarly, we can write specs (test cases) for other components as well