**T14-HOL001\_TestCase[Understanding the e2e testing with Protractor]**

Here we will understand the file structure for performing e2e testing with Protractor and how to execute the test. Also we will be understanding how to capture the Protractor Test Result. Protractor is used to perform end to end testing for Angular applications. Protractor is a Node application written on top of Selenium’s WebDriverJS.

**Tasks:**

1. Create an angular application called T14-HOL001
2. Expand the e2e folder
3. Check the **protractor.config.js** file. This file contains the default configuration information to run end-to-end testing using Protractor [which you can change/configure depending on the necessity of your application].

**Note:**

* ‘browserName’: ‘chrome’ is used to indicate that Chrome will be used as the browser to execute the tests.
* directConnect: true allows Protractor to directly connect to the browser and instead of going through a Selenium Standalone Server. When you have this configuration in place, you don’t have to manually start a separate Selenium Server.

1. Check the **tsconfig.e2e.json file**. This file tells the transpiler to emit ES5 JavaScript, and that the JavaScript output files should be written to out-tcs/app directory.
2. Check the **app.po.ts** file.

**Note:**

* Page Object Model is a design pattern which is commonly used in test automation circles. A page object represents a page or a portion of an application using a class. The sole purpose of this class is to capture different elements (headings, tables, lists, buttons, links and etc) in the page and optionally perform tasks (clicking buttons, setting text on input elements, retrieving text) on them. We can then import these page objects into the spec file and invoke their methods. This reduces code duplication and makes maintenance of code easier.
* The sample PO class created for app page by the Angular CLI contains two methods, namely navigateTo and getTitleText . navigateTo method is used to enter a URL in the browser and navigate to a web page. getTitleText method is used find the h1 element located under app-root element and extract its text value.

1. Check the **app.e2e-spec.ts** file. Spec files contain the tests and the tests are written using the Jasmine framework. In this application, app.e2e-spec.ts contains only one test. It basically navigates to http://localhost:4200 , extract the title and checks whether the its value equals to Welcome to T14-HOL001!.
2. Now to execute the e2e test using protractor run the command:

**ng e2e –port=4300**

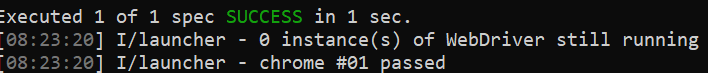
***[Note: Here we are telling Protractor to use a different port other than default 4200 port because by default the 4200 port is used for application execution. So 4200 may be busy and hence may not execute your protractor tests. You can give any free port number of your choice].***

**Sample Output:**

Once your run the ng e2e command you can see the web driver for chome will be downloaded [as you have defined chrome as default browser to run automated test in protractor.config.js file].

****

The tests will give the output as successful if test cases pass:



Also as we are running automated test, the chrome will be opened and closed by default to display test output of executed test. To capture the screenshot of your protractor test result, do the following:

1. Run the command: **npm install screenshot-protractor –save**
2. Then under the **app.e2e-spec.ts** file add the below command:

afterEach(() => {

browser.takeScreenshot().then((png) =>{

var stream = createWriteStream("samplescreenshot.png"); /\*\* change the png file name as per your choice  \*/

stream.write(new Buffer(png, 'base64'));

stream.end;

});

});

Now when you run the command ng e2e the screenshot of the protractor test result will be captured and saved under your application as given below in the sample:

