Effective AngularJS Web Testing using Protractor

Deakin University

Facuty of Science, Engineering and Build Environment

Assignment 1

Tharini Hasara Manawadu De Silva – 217336638

Executive Summery

Software Testing, an essential feature in software development industry to deliver high quality software product to the customer. In this paper is an overview of analysis and evaluation of threading software automation tools and testing methods. This analysis is basically on AngularJS test automation tools and software testing methods to use to perform on an AngularJS based web application. It define some testing tools as well as their functionalities and then correctly choose an appropriate testing tool for selected web application. Then it is discussed the testing methods which suitable for the web application. Result of this analysis, Protractor was selected as the best testing tool for AngularJS based web applications when considering its specialties on end-to-end AngularJS automation testing. This report finalises that how Protractor tool can be used to test Angular based web applications and how it performs end-to-end and unit testing on that particular web application.

Recommendations discussed include:

- Improving efficiency and effectiveness of software testing tools.
- Trending JavaScript base web application development.
- Enhancing AngularJS based automation testing functionalities.
- More utilization on testing methods.

This paper has some limitations. Some of these limitations are:

This research conducted basically around a web application. Tools and testing methods used here are selected prior to that web application. Since the selected web application is an AngularJS based web application, the selected tools are basically JavaScript supported automation tools.

Introduction

Modern web applications, sites highly focus on effectiveness, efficiency, high performance, reusable components and user friendliness unlike traditional web sites. Client side technologies and functionalities save bandwidth while on transactions and minimize the occurrence of overhead in the server, this delivers the web application functionalities faster. AngularJS is a JavaScript framework that can be used to reach the above mentioned modern web application characteristics, also it has become a trend because it ease off the development process. AngularJS extends HTML properties by using Angular Directives to create rich internet application. The simplicity of data binding, templating, routing, animating and validating of AngularJS it is applied for Single Page Application development with ASP.NET, MVC.

AngularJS is used to create broad scale and high performance web sites and applications while continuing the maintainability. Most of the web applications with complex functionalities have been developed using AngularJS, once such e-commerce web application is LANDS' END (www.landsend.com) that used as the program throughout this research. LANDS' END is an online shopping mall which facilitates purchasing clothes and house hold items.

Delivering a quality software product is the crucial point in software development process. Even it used the best development technologies, to maintain the quality, integrity, efficiency and effectiveness of the product, the process of software quality testing is essential before handover the software product to the customer. To get the best output in a web application, it must be test, whether the web application functions as expected. So as testing methods, functional testing and unit

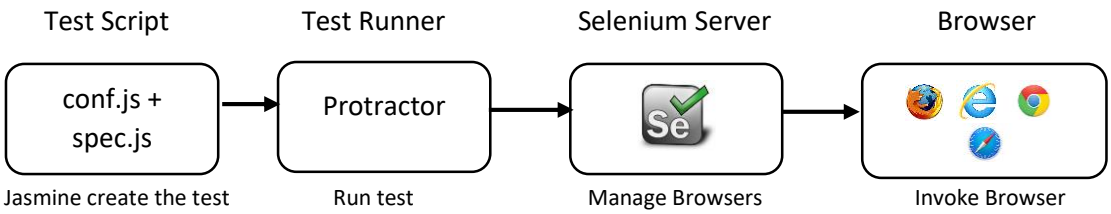
testing are covered in this research to test whether the LANDS' END online shopping mall behave as expected.

Test a large scale web application manually is time consuming and costly. Use of test automation tools for testing, automatically verify main functionalities, help to perform on huge number of test functions in a short time period and test for regression. So automated software testing in software projects becomes more important. AngularJS web application testing cannot be performed by traditional testing automation tools, because AngularJS directives define like ng-repeater, ng-model, ng-controller… etc., which are not included in other automation tool locators. The intention of AngularJS base automation testing can satisfied only by using a specific test automation tool like Protractor that support AngularJS directives.

Protractor is an end-to-end automation testing framework that specifically used for AngularJS web applications and sites testing. Protractor is a node.js framework and runs on Selenium 2.0/WebDriver. It functions as an abstraction layer on Selenium Server and allows to write test automation, then invokes for AngularJS web applications and sites. Protractor inherited all the properties and behaviours from Selenium2.0 and also added AngularJS specific commands, services and functions to matching with AngularJS web application testing. Technologies like Jasmine, Cucumber and Mocha have combined with Protractor to give an integrated solution. Protractor use Jasmine to create the test. Jasmine is a behaviour driven development (BDD) framework which underlying JavaScript and node.js. Jasmine provide syntax, model and reporting tools to proceed the testing process.

Protractor optimizes efficiency of test automation process by avoiding waiting and sleeping times. It facilitates both real browser and headless browser testing. Protractor highly supports for functional and unit testing that basically focused in this research.

How Protractor works



Literature Review

Protractor is a framework that is used to end-to-end testing in AngularJS applications and web sites. Protractor is used for functional and unit testing. The main purpose of Protractor functional testing is fulfill the acceptance criteria requested by the customer, not only testing the AngularJS application. WebDriverJS environment facilitates Protractor to be built. It concatenates user application and browser-specific drivers. Protractor allows to process locator strategies which specific to AngularJS. These strategies helps test AngularJS elements without a serious configuration effort. Protractor performs an asynchronous testing process. The tester no longer need to insert wait and sleep times to the test case. Protractor starts automatically the next test scenario while executing the current test and no need to concern about web synchronous.

Selenium runs on Selenium server and inherits all the advantages and values from Selenium. Since Protractor build on Selenium server, it can access WebDriver protocols like ChromeDriver and GhostProtocol. Selenium server perform on interpreting testing commands come from test and pass

them to browsers (https://docs.angularjs.org/guide/e2e-testing). The communication among server and browsers is done using WebDriver's Wire protocol which is a JSON type protocol. Selenium WebDriver functions as an object oriented API for access the DOM (Document Object Model) of a web application. It helps test script development write in a high-level language. Automatically run those scripts give very fast and detailed feedback to the web tester (Stocco et al. 2014). Selenium WebDriver gives quick and well-structured behavior and easy maintenance automated test scripts. It supports to divide the test project into technical parts and business test parts (Bures & Filipsky 2016).

The invention of Selenium made a revolution in Software Testing Industry. Selenium started to act a major role in web application and web site testing. Selenium was invented accidently, while processing a software application testing manually. To avoid the repetitive testing on same steps an automation process was created using a JavaScript library which became the core of Selenium. It is a free, open source testing tool (Altaf et al. 2015). Most of the instances of in-page JavaScript could be handled by Selenium. Selenium uses JavaScript and iframe to concatenate test automation engine and the web browser. This facilitates reusability of scripts in different browsers in various platforms. In each iteration of agile process, Selenium test build creates a Regression Test Suite in web tier of the application. This allows developers to change the view without breaking the previously passed acceptance test. Selenium test could be written without a deep understanding of implementation, because test shell of Selenium is very quick (Holmes & Kellogg 2006). Selenium server communicates with the test project in a simple way. First the server receives commands from test project, then it translates the request and return back to the test project. Every HTTP GET/POST request could be utilized through Selenium server (Altaf et al. 2015).

Even though Selenium is a leading and considered as the best testing tool for web application testing, it doesn't support AngularJS directives while Protractor supports them. As examples ng-repeat, ng-model, ng-controller, ng-app are some of them. Protractor facilitates novel locator strategies and methods like waitForAngular, By.repeater, By.binding, By.textarea, By.model, WebElement.evaluate, etc., to make ease the automation of AngularJS applications which is not supported by Selenium. Protractor facilitates to use global elements, which get locator parameters and return an Element Finder. This contains some action methods like click(), sendKeys(), getText(), perform(), etc., which is the foundation to connect with elements and extract information. The below code examples show that how Protractor reduce the syntax complexity on element locating with compare to Selenium.

Using Selenium

```
14
15  driver.findElement(By.cssSelector("css selector"));
16
```

Using Protractor

```
14
15  element(by.css('some-css'));
16
```

Protractor is an e2e test automation framework for AngularJS applications while Selenium is a script recording and playback tool. (Ebook-agile-software-testing.pdf). It's not a difficult or serious thing that selecting a tool between Protractor and Selenium for a testing. When the web application is developed using AngularJS, reactJS or emberJS, it is recommended that to be used Protractor which provided more advantages than Selenium. Both Selenium and Protractor are constructed on WebDriver with sort of differences. Here it is considering about MVC (Figure I) and MVVM (Figure II) model. Generally AngularJS applications enabled asynchronous, which means that model is updated automatically

when view get updated and vice versa. So that means Protractor supports asynchronous calls and makes application more efficient. Protractor avoid sleep and wait time by asynchronous testing process while Selenium occupy much sleep and wait time because of the synchronous testing process.

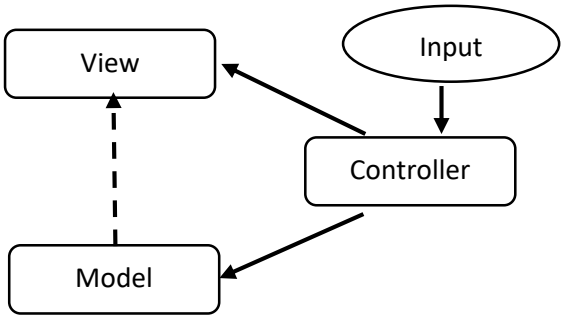


Figure I

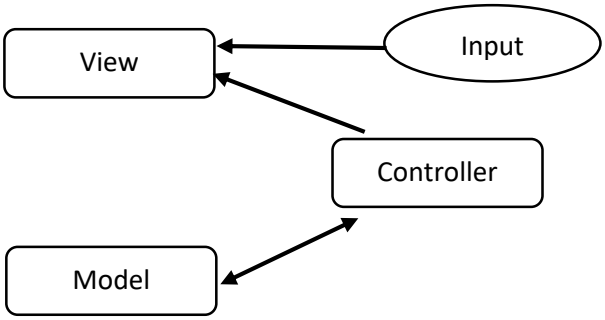


Figure II

Karma is a web automation testing tool which also could be used for AngularJS web application testing. Karma is created in JavaScript. NodeJS is used to process server part while web browser is used to run the client. The client applications communicate with server using socket.io library and test is run on iframe. Socket.io facilitates a function based communication channel. Server blows functions while client is listening and vice versa. Multiple protocols like WebSocket, HTML polling, JSONP polling and XHR polling are implemented by socket.io and uses the best web browser available (Dutta, Verma & Prithviraj 2015).

The Karma test runner which was built by Angular team inspects the test files and run the test at any time that needed. If any file is get changed, it triggers the test execution by giving a signal to the test server. Karma test runner executes the unit test inside the web browser and shows the test results. Karma allows for parallel browser processing and proved that AngularJS web application executed as expected (Knol 2013). Karma supports many other testing frameworks like Jasmine, Mocha, Qunit. In Karma, each and every step of testing need preparation. Services, dependencies and filters should be initialized before the testing process. After that, it is generated test specification. Then the result expression compare with matcher expression and should be equal with each other. Matcher expressions are usually handled in expect functions. These are different type of expressions like string, Boolean, etc. (Fat et al. 2016).

Even Karma supports AngularJS web application testing, it only provides unit testing features while Protractor supports both unit and end-to-end automation testing. Unit testing is a good approach that ensure the security of the application after modifying the code. However, unit testing does not fulfill the entire application testing (Gharat & Nehlsen 2014). Karma is a test runner, not a test framework while Protractor is a test framework. Karma just runs the test and let know the results whether pass or fail, but a test framework write test rather than just run it. (AngularJS test driven development book) Karma is focused on Test Driven Development (TDD) while Protractor is focused on Behavior Driven Development (BDD) which includes TDD. Karma is not supported for server side test which used HTTP GET, POST request methods the application uses NodeJS while Protractor supports server side test (Fat et al. 2016).

Jasmine is a Behavior Driven Development test framework which is specified for test JavaScript source codes. It has no any dependency with other JavaScript frameworks. Jasmine does not need a DOM

model to process and it makes easy to write test with its simple syntax. It allows to execute browser tests and NodeJS tests in same environment (https://jasmine.github.io/). It was implemented using Eclipse. It began as a Java application and has created in Eclipse plugins. Jasmine can reach the model, source code and test in same platform. It has integrated model-level integration, source-level invoke and final verification (Zhou et al. 2008). Jasmine is a Behavior Driven Development (BDD) framework for testing only JavaScript source code while Protractor is also a BDD framework which focused on end-to-end testing. Jasmine only test JavaScript code while Protractor focuses on functional testing (Karttunen 2016).

Nightwatch.js is a NodeJS based end-to-end testing framework. It runs on Selenium and aim is to functioning web browser automation testing. It has simple and powerful syntax to write tests only using JavaScript (NodeJS) and Xpath or CSS selectors. To locate or verifying the elements in the page or invoke commands, it can only use Xpath or CSS selectors. To invoke commands and assertions on DOM model elements, it uses W3C WebDriver API (http://nightwatchjs.org/). Architecture of the NightWatch is an implementation of W3C WebDriver which is a JSON wire protocol while Protractor runs on Selenium WebDriver by inheriting all the features of Selenium WebDriver. NightWatch is not supported Angular specific locator strategies while Protractor fully supported. Protractor has in-built support to Typescript, but NightWatch not support to Typescript.

WebdriverIO is a JavaScript based web automation tool which focuses on both BDD and TDD test frameworks in JavaScript world. It is a W3C WebDriver JSON wire protocol. It simply integrated with other tools to enhance testing features and it supports for all WebDriver protocol commands (http://webdriver.io/). When comparing WebdriverIO with Protractor, WebdriverIO partially support AngularJS specific locator strategies. It only implemented WaitForAngular locator to support AngularJS directives.

According to the above analysis with different software automation testing tools and frameworks, It can be claimed that Protractor is the best software automation tool for AngularJS base web automation testing.

Evaluation and justification

The web application (LANDS' END) which is used to test has built using AngularJS. Nowadays AngularJS is famous for developing web applications, because of its extended functionalities in HTML, easy to code and quick to market. In my quest to find a better functional and unit automation tool for AngularJS based web application to automate the functional and the UI of the web application, I finally decided to select the Protractor automation tool.

LANDS' END is a huge e-commerce web application which facilitate users to perform online shopping. Testing a large scale web application like LANDS' END manually is time consuming and costly. So automating the web application build in AngularJS by using Protractor is time effective and reduce the cost. Since LANDS' END is an AngularJS based application, Protractor facilitate novel locator Strategies and methods to automating the test process of the web application. When examine the source of the LANDS' END it uses AngularJS directives like ng-repeat, ng-controller, ng-class, ng-hide, ng-if, etc. AngularJS based testing tool like Protractor can clearly locate these directives. Protractor with the collaboration of Selenium render the infrastructure for the test automation take the full use of Selenium to parallel browser execution. Automating LANDS' END through Protractor makes the testing process more comfortable in creating and executing test automation scripts.

Basically as testing methods, functional testing and unit testing are covered in this research to test whether the LANDS' END e-commerce web application behave as expected. Protractor supports end-to-end testing as well as unit testing. With the use of Protractor it make easy the functional testing and unit testing process by using same language and same framework to write tests for both test methods. The main difference of these two testing methods is that part of the web application that the testing methods are allow access.

Model and the controller can be accessed by unit test. The tools which are used to unit testing can inject filters, services and controllers that facilitate tester to test each and every module of the web application separately (Fat et al. 2016). Unit testing complements the system level testing and integration testing. Generally unit testing consider as a white box testing. It focusing and evaluating code, as it implemented rather than focusing to requirements (Parkin & Australia 1997).

End-to-end test only have permission to access the view. It can test main functionalities like opening a new view after button was clicked. End-to-end application testing make sure that overall functions of the application work as expected (Fat et al. 2016). End-to-end (E2E) testing is different from unit testing which focuses only on single module testing. E2E testing is very similar to integration testing, but E2E testing focuses from end user's point of view and integration testing focuses on any subset of subsystem. In E2E testing it specifically find the requirements of the system under test scenario, then generates test cases based on the required specifications (Tsai et al. 2001).

In this research, I highly focuses on Functional testing and then focuses on unit testing, because the web application that is chosen for this research is basically need to test whether the web application functions as expected and also check whether it gives proper end user support. Then the web application need some unit testing on some modules like searching for a good, adding to the shopping cart, etc. Then it is easy to process the overall testing effective and efficient way.

Conclusion

The process of executing a program or application with a strong desire to discovering software errors or issues, define as software testing and this effort verify that software product is ready to use. Software testing is an integral step in software development life cycle to deliver efficient and effective product. According to this paper it shows that the best automation tool for end to end testing for AngularJS web applications is Protractor. Whether Protractor runs on top of Selenium, Protractor shows specific functionalities beyond Selenium in AngularJS web automation testing. Although Karma, Jasmine, WebdriverIO and NightWatch are JavaScript based testing tool, only Karma supported AngularJS web automation testing. When compare Karma and Protractor, Karma only focused on Test Driven Development (TDD) while Protractor is focused on Behavior Driven Development (BDD) which includes TDD. As a conclusion, it is beneficial to use Protractor on AngularJS based end-to-end web automation testing. Automate testing for web applications build in AngularJS with the use of Protractor is a crowing step ahead for software development industry in the effort of automating business process applications.

Bibliography

Altaf, I, Dar, JA, ul Rashid, F & Rafiq, M 2015, 'Survey on selenium tool in software testing', in *Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on*, pp. 1378-83.

Bures, M & Filipsky, M 2016, 'SmartDriver: Extension of Selenium WebDriver to Create More Efficient Automated Tests', in *IT Convergence and Security (ICITCS), 2016 6th International Conference on*, pp. 1-4.

Dutta, P, Verma, A & Prithviraj, J 2015, 'Karma-The Test Runner, for Automated Testing of Web Based Applications', *International Journal of Innovative Research in Science & Engineering*, vol. 3, pp. 31-7.

Fat, N, Vujovic, M, Papp, I & Novak, S 2016, 'Comparison of AngularJS framework testing tools', in *Zooming Innovation in Consumer Electronics International Conference (ZINC), 2016*, pp. 76-9.

Gharat, A & Nehlsen, M 2014, *AngularJS UI Development*, Packt Publishing.

Holmes, A & Kellogg, M 2006, 'Automating functional tests using selenium', in *Agile Conference, 2006*, pp. 6 pp.-275.

http://nightwatchjs.org/ *Nightwatch.js*.

http://webdriver.io/ *WEBDRIVERIO*.

https://docs.angularjs.org/guide/e2e-testing *ANGULARJS*.

https://jasmine.github.io/ *Jasmine Behavior-Driven Javascript*.

Karttunen, J 2016, 'Unit testing of AngularJS: A look into writing tests for web application'.

Knol, A 2013, *Dependency Injection with AngularJS*, Packt Publishing Ltd.

Parkin, R & Australia, I 1997, 'Software Unit Testing', *The Independent Software Testing Specialists, IV & V Australia*.

Stocco, A, Leotta, M, Ricca, F & Tonella, P 2014, 'PESTO: A tool for migrating DOM-based to visual web tests', in *Source Code Analysis and Manipulation (SCAM), 2014 IEEE 14th International Working Conference on*, pp. 65-70.

Tsai, W-T, Bai, X, Paul, R, Shao, W & Agarwal, V 2001, 'End-to-end integration testing design', in *Computer Software and Applications Conference, 2001. COMPSAC 2001. 25th Annual International*, pp. 166-71.

Zhou, Z, Wang, L, Cui, Z, Chen, X & Zhao, J 2008, 'Jasmine: A tool for model-driven runtime verification with UML behavioral models', in *High Assurance Systems Engineering Symposium, 2008. HASE 2008. 11th IEEE*, pp. 487-90.