International Journal of Engineering Technology and Applied Science
(ISSN: 2395 3853), Vol. 2 Issue 7 July 2016

Implementation of Page Object Model in Automation script for Sencha UI using Selenium

[1]Sumit Sabu, [2]Mrs. Priyanka Dhasal

[1] M. Tech. Student – 4th Semester- Information Technology Branch, Patel College of Science & Technology (PCST), Ralamandal Indore, Madhya Pradesh 452020

[2]Assistant Professor - Information Technology Branch, Patel College of Science & Technology (PCST), Ralamandal Indore, Madhya Pradesh 452020

ABSTRACT:

Every software development group tests its products, and then too delivered software always has defects. Test engineers strive to catch them before the product is released but they always creep in and they often reappear, even with the best manual testing processes.

Test Automation software is the best way to increase the effectiveness, efficiency and coverage of your software testing.

Manual software testing is performed by a human sitting in front of a computer carefully going through application screens, trying various usage and input combinations, comparing the results to the expected behavior and recording their observations. Manual tests are repeated often during development cycles for source code changes and other situations like multiple operating environments and hardware configurations. An automated testing tool is able to playback pre-recorded and predefined actions compare the results to the expected behavior and report the success or failure of these manual tests to a test engineer. [1] Once automated tests are created they can easily be repeated and they can be extended to perform tasks impossible with manual testing.

Current automation scripts are designed to locate static HTML elements, but this script not only identify the static elements while also been able to detect the Dynamically generated elements (for eg- [2] Sencha UI elements occupy different ID when page reloads or user navigates to another page). This script will help the user to perform repetitive regression testing task with much more less effort and minimum amount of time.

KEYWORDS:

Sencha Automation, Automation script, Selenium script for sencha, Dynamic UI Element Automation

INTRODUCTION:

Test automation has been quite hot topic for multiple years. There are many problems in that field. Some people are seeing it as "total no no no - just waste of money and time", some people think it is the silver bullet, which solves all problems. My view is somewhere between those two extremes. One important aspect of making good test automation is to understand when to automate, how to automate and why to automate. It's more process related than implementation related. So here are some questions - Why do you automate? How do you decide what to automate or not? Does it help to make you test automation better and cheaper? What kind of risks there is? What kind of quality aspects your test automation has? While automating any web application using an automation tool like selenium web driver. We have to identify locators for elements which we need to interact with. It could be ID, Name, CSS Selector, XPath or combination of all these. It is quite straightforward to identify locators for static elements which are clearly defined with static IDs. But in some applications, we come across dynamic elements and it becomes quite challenging to identify locators for such dynamic elements. Dynamic elements are those elements which have identifiers that are dynamically generated. Dynamic identifiers are normally used for buttons, text-fields and text-areas etc. The main purpose of creating the dynamic elements is to protect the application from various malicious attacks like Brute force attack (in which

some important modules of any application are maliciously attacked by automated scripts against the security, performance or stability issues). So many developers nowadays decide to use the architecture of the dynamic elements. By using the current automation script, users are now able to automate the dynamic elements created in Sencha Touch (It is a user interface (UI) JavaScript library, or framework, specifically built for the Mobile Web. It can be used by Web developers to develop user interfaces for mobile web applications that look and feel like native applications on supported mobile devices).

PROPOSED ARCHITECTURE AND TECHNIQUE:

The architecture of current automation script is POM (Page Object Model) and Page Factory architecture. A Page Object Model is a design pattern that can be implemented using selenium web driver. It essentially models the pages/screen of the application as objects called Page Objects, all the functions that can be performed in the specific page are encapsulated in the page object of that screen. [3] [4] In this way any change made in the UI will only affect that screens page object class thus abstracting the changes from the test classes.

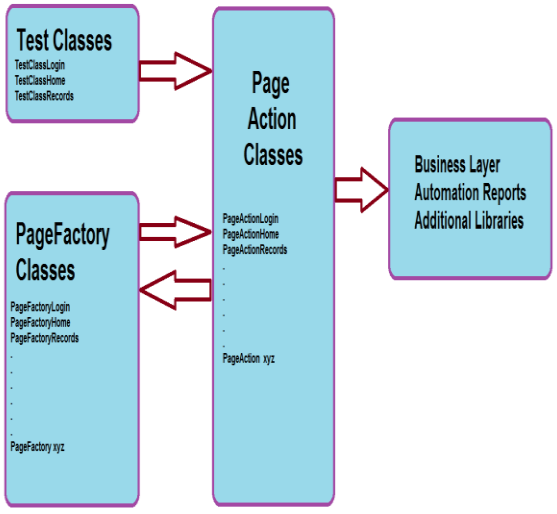

Fig 1: Architecture Design

Advantages of POM:

1. **Object Repository**: You can create an Object Repository of the fields segmented page-wise. This as a result provides a Page Repository of the application as well. Each page will be defined as a java class. All the fields in the page will be defined in an interface as members. The class will then implement the interface.
2. **Functional Encapsulation**: All possible functionality or operations that can be performed on a page can be defined and contained within the same class created for each page. This allows for clear definition and scope of each page's functionality.
3. **Low maintenance**: Any User Interface changes can swiftly be implemented into the interface as well as class.
4. **Programmer Friendly**: Robust and more readable. The Object-oriented approach makes the framework programmer friendly.
5. **Low Redundancy**: Helps reduce duplication of code. If the architecture is correctly and sufficiently defined, the POM gets more done in less code.
6. **Efficient & Scalable**: Faster than other keyword-driven/data-driven approaches where Excel sheets are to be read/written.

The main advantage of Page Object Model is that if the UI changes for any page, it don't require us to change any tests, we just need to change only the code within the page objects (Only at one place). Many other tools which are using selenium, are following the page object model.

ALGORITHM:

1. Create page objects for all available pages in our application.
2. In those pages we will try to get the handle for the elements present on the corresponding pages using PageFactory architecture [5].
3. We will need to apply different approaches to extract the element. (As Sencha-JavaScript UI framework, provides dynamic elements).
4. Create the action classes for each page of the application. These action classes contain various actions that needed to apply at the different elements at different times.
5. Create test classes for each individual page present in the application. These classes contain the methods, which are responsible for invoking the action methods of the corresponding pages.
6. Screenshots of the desired screen and condition (either Fail or Pass) with suitable name and location is taken and saved to the directory.
7. Generate a test report.

EXPECTED OUTCOME:

After the implementation of this script of selenium for Sencha UI elements, users will be able to handle most of the dynamic elements present on any webpage or website. This script also takes the screenshot (at desired conditions) of the desired pages at any particular time (depends on the requirements/need of that).This automation script of the web pages helps users in regression testing and retesting tasks. [6]s Usually these tasks take so much manual effort and time. By using this automation script the repetitive tasks are now been scripted into the code and user can easily check the particular area of the site without putting so much time or effort.

CONCLUSION AND FUTURE WORK:

Testing is the process of evaluating a system or its component(s) with the intent to find that whether it satisfies the specified requirements or not. This activity results in the actual, expected and difference between their results. In simple words testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements.

The Automation Testing is used to rerun the test scenarios that were performed manually, quickly and repeatedly. Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms. Page Factory is an inbuilt page object model concept for Selenium Web Driver, but it is much optimized. Page Factory can be used in any kind of framework such as Data Driven, Modular or Keyword Driven. Page Factory gives more focus on how the code is being structured to get the best benefit out of it.

Often developers uses these type of dynamic element framework in order to protect the website/application from various malicious attacks like Brute force attack, security attacks etc. This automation script is providing a demonstration of how the dynamic elements are automated and use can perform regression test on those application.

REFERENCES:

[1]. http://www.ijarcce.com/upload/2016/may-16/IJARCCE%20105.pdf
[2]. https://www.seleniumeasy.com/selenium-tutorials/test-automation-scripts-structure
[3].http://www.sciencedirect.com/science/article/pii/S1877050915005396
[4].http://www.ijcsit.com/docs/Volume%205/vol5issue01/ijcsit20140501196.pdf
[5]. http://eliga.fi/Thesis-Pekka-Laukkanen.pdf
[6]. http://www.guru99.com/page-object-model-pom-page-factory-in-selenium-ultimate-guide.html