****

**Selenium Automation Framework**

**Introduction –**

An Automation Framework is defined as a set of abstract concepts, processes, procedures and environment in which automated test cases will be designed, created and implemented. It includes physical structures used for test and implementation, as well as the logical interactions among these components.

We have created Java based Web Automation Framework built on top of Selenium, Maven and TestNG.

The flexibility of this framework is that it allows you to execute test cases either from eclipse or from Batch files with the help of Maven build or even via Jenkins Job. This helps an external user who has very little knowledge on Selenium to execute test case by simple few clicks.

**Key Tools/Technologies –**

1. Selenium Web Driver
2. Java
3. TestNG
4. Maven

**Framework Features –**

* Data Driven Support using Excel –
* You can fetch data easily from excel and pass that to test case. We have used POI library to connect to excel.
* Framework will read data from given excel file and convert test case into data driven test case.
* Bug Tracking Tool Integration –
* Framework will automatically log a defect for all failed test cases with screenshot and other browser details.
* Currently we have integrated JIRA. Similarly, we can integrate other defect tracking tools as well.
* Database Integration –
  + Framework gives you flexibility where you can verify or compare values between UI and DB.
* Test Management Tool Integration –
* Page Object Model –
* Framework provides Page Object Model support.
* You can write your scripts in Page driven manner.
* Parent Child Component Structure – Framework is created as per the hierarchy of parent child component.
* Built In Wait Methods –
* Now you don’t need to worry about wait and AJAX calls. Framework provides more than 10 wait methods that you can use in your scripts.
* Rerun Failed Test Cases –
* Framework will return failed method name in test execution report along with line number.
* Screenshot on Failure –
* Framework has the ability to take screenshot for every test on failure.
* Support Headless Testing –
* Framework has in-built support to headless browser testing to speed up you test execution.
* Capable of Integrating with Continuous Integration Server like Jenkins –
* We have integrated Jenkins war file in framework itself.
* We have created and integrated corresponding batch file for Jenkins war file so that a user that has no knowledge on how to work with Jenkins can get related tasks done by simply double clicking on Jenkins batch file.
* Capable of Parallel Testing –
* With the help of framework, you can execute you tests in parallel on different browsers & machines.

**Framework Architecture –**

* Listeners –
* TestNG Listeners
* Suite Listeners
* Data Providers –
* Excel Data Providers
* SQL Data Providers
* String Utility –
  + Framework is loaded with several in built methods related to String like generate random string.
* Number Utility –
  + Framework is loaded with several in built methods related to String like generate random number.
* You can fetch data from excel, database and text files (config/property files) to create data driven test cases.
* User can execute test cases in 3 ways. Command Prompt, Eclipse and Jenkins. For command prompt, you need to execute equivalent Batch files. For Eclipse, you need to run Configuration files in config folder after setting up run configurations as TestNG, Ant or Maven depending on the nature of your project. For Jenkins, you need to create corresponding Jenkins Job.
* Execution can be done locally or in parallel way on different machines.
* Finally, user will get HTML/XML Test Execution reports and Test Execution Logs.

**Framework Project Folder Structure –**

**References –**

For complete reference on QA Fiction Selenium Java Framework, follow below link –

<https://qafictionblog.wordpress.com/>