**Selenium Automation Framework**

Implementing Automation on any Web Based Application using Selenium is no more a challenge now!! Just extend this Framework by including it in your build path & leverage the reusable components which would help developing Automated Test Scripts for your application very easy and much faster...

**Framework Features:**

* It’s a Generalized, Platform independent, Scalable and Extensible framework for automated testing of Web Applications.
* The Framework has been built up using open source tools Selenium WebDriver, Selenium Grid, ReportNG, TestNG, Maven & Jenkins.
* Supports Parallel execution of Tests on Multiple Browsers and Multiple Machines that reduces execution time.
* Automated Re-run of failed Scripts.
* ActionMethods Library with a set of 100+ reusable actions which can’t be directly performed using Selenium.
* Easily Integrable with Jenkins to support CI and Scheduled runs
* HTML reports with failure Screenshots.
* Grouping of Test Scenarios and selective execution.
* Uses open source tools.
* Dynamic object locators (e.g. clickAndWait(ResourceCenterPage.Filter\_Option.format("Client Stories"),2);)
* Implements standard Data driven and Keyword Driven approaches in Automation.
* Implements Page Object Model
* Platform Independent. ( Windows, Mac OSx, Linux)
* Supports AJAX based Web applications.
* Multiple browser support. (Firefox, IE, Chrome, Opera, Safari)

**How to Use:**

* Clone the project in Eclipse
* Create pages specific to your application in "pages" package in main and define objects and page specific methods in it.
* Create factory methods for each page in PageFactory class in global package. e.g. included
* Define Test Data in BaseData class in data package
* Write your tests in tests package in test by referring to the sample Tests included. While writing tests, Reuse the existing 100+ set of re-usable actions with java-docs defined in ActionMethods class. This actions have implicit reporting implemented for failures.
* Configure testng.xml with the test groups to run and parallel thread count
* Configure test.properties with the browser, server.. on which to run the tests
* Start the Selenium stand alone server using command > java -jar selenium-server-standalone\*.jar
* Run the tests by right click > Run As > TestNG Suite option on testng.xml
* Check the Test Reports by opening test-output/html/index.html folder