Automation testing notes

esi.activity

Muhammad U. Bari

2015

# Resources

## Selenium tutorial ([link](http://www.toolsqa.com/selenium-webdriver/selenium-tutorial-selenium-webdriver-tutorials/))

* JavaDocs
  + Selenium API [JavaDocs](http://seleniumhq.github.io/selenium/docs/api/java/index.html),
  + Apache Commons: [JavaDocs](https://commons.apache.org/proper/commons-io/apidocs/index.html)
  + JUnit: [JavaDocs](http://junit.org/javadoc/latest/)
* Selenium cheat sheet: [link](https://gist.github.com/huangzhichong/3284966),
* Selenium xPath [tutorial](http://www.toolsqa.com/selenium-webdriver/choosing-effective-xpath/)
* Selenium: finding elements [tutorial](http://www.toolsqa.com/selenium-webdriver/findelement-and-findelements-command/) ,
* Finding elements using CSS: [link](http://stackoverflow.com/questions/27243184/get-element-by-cssselector-in-selenium-java),
* Slowing down selenium: The exported code runs too quickly on java and causes the test to fail because it can’t find the element.
  + Overriding findElement method: [link](http://stackoverflow.com/questions/32067341/set-selenium-webdrivers-default-execution-speed)
  + Using Thread.sleep command before executing Selenium methods. (Not preferable but easy)
* Selenium: [file upload](http://stackoverflow.com/questions/9431978/one-solution-for-file-upload-using-selenium-webdriver-with-java)
* Taking screenshot with selenium: [link](https://loadfocus.com/blog/2014/06/04/how-to-take-snapshots-of-the-browser-using-selenium-webdriver/),

## Miscellaneous

* Bypassing Firefox HTTP authentication: [link](http://stackoverflow.com/questions/3021602/http-basic-auth-via-url-in-firefox-does-not-work),
* Setting values on Java enum: [link](http://stackoverflow.com/questions/8811815/is-it-possible-to-assign-numeric-value-to-an-enum-in-java),
* Attaching Javadocs to eclipse user libraries: [link](http://stackoverflow.com/questions/9873152/how-to-attach-javadoc-or-sources-to-jars-in-libs-folder),
* Formatting numbers: [link](http://stackoverflow.com/questions/3672731/how-can-i-format-a-string-number-to-have-commas-and-round), [link](http://stackoverflow.com/questions/17296691/how-to-format-a-double-with-variable-amount-of-decimal-digits),
* Formatting dates: [link](https://docs.oracle.com/javase/tutorial/i18n/format/simpleDateFormat.html),
* Using Git in ANT: [link](http://tlrobinson.net/blog/2008/11/ant-tasks-for-git/),

## JUnit ([link](http://junit.org/javadoc/latest/))

* Junit parameters: [link](https://github.com/junit-team/junit/wiki/Parameterized-tests),
  + Used for executing each test case by number of browsers available.

## Apache ANT (Another Neat Tool) ([link](http://ant.apache.org/))

* Apache ANT (Another Neat Tool): [tutorial](http://www.vogella.com/tutorials/ApacheAnt/article.html), [link](http://examples.javacodegeeks.com/core-java/apache/ant/building-java-application-with-ant-and-eclipse-example/),
  + Installing ANT: [link](https://www.youtube.com/watch?v=arTLYV3_po4),
  + Getting the current date: [link](http://www.asjava.com/ant/ant-how-to-echo-current-time-and-system-date-in-ant/),
  + Running JavaDocs from ANT: [link](https://ant.apache.org/manual/Tasks/javadoc.html), [link](http://stackoverflow.com/questions/1495982/how-to-generate-javadoc-with-ant-for-an-existing-project),
  + Passing argument to testNG:, [link](https://ideoplex.com/2003/07/11/setting-java-system-properties-with-ant/), [link](http://stackoverflow.com/questions/6776135/how-to-pass-parameters-for-a-ant-script-which-is-invoked-via-shell-script), [link](http://stackoverflow.com/questions/3937817/how-to-read-in-arguments-passed-via-ant-to-testng-xml),

## Apache POI

* Used to read excel files.
* Reading a date from excel file: [link](http://stackoverflow.com/questions/3148535/how-to-read-excel-cell-having-date-with-apache-poi),
* Checking if cell is empty: [link](http://stackoverflow.com/questions/15764417/how-to-check-if-an-excel-cell-is-empty-using-poi),

## TestNG ([link](http://testng.org/doc/index.html))

* TestNG
  + General documentation: [link](http://testng.org/doc/documentation-main.html),
  + JavaDocs: [link](http://testng.org/javadocs/index.html),
  + Running multiple test cases: [link](http://www.mkyong.com/unittest/testng-tutorial-5-suite-test/),
  + Adding logs: [link](http://testng.org/javadoc/org/testng/Reporter.html),
  + Running TestNG from command line: [link](http://testng.org/doc/documentation-main.html#running-testng),
    - Specifying the output folder (for report) in command line (): -d <directory to be placed for output folder>
      * -d is an argument. Make sure the proper command is called to run TestNG
  + Data provider: [link](http://examples.javacodegeeks.com/enterprise-java/testng/testng-dataprovider-example/),
    - Used in order to run the test cases on different browsers.
  + Adding colour to reporter log: [link](https://groups.google.com/forum/#!topic/testng-users/jEhDFx7Jr9Q),

## HighCharts ([link](http://www.highcharts.com/))

* Used to generate graphs
* Extracting graph information
  + Code was produced to iterate through the graphs. Located on [github](https://github.com/Ardesco/Powder-Monkey/tree/master/src/main/java/com/lazerycode/selenium/graphs), the testing framework uses refactored code of the one from github

# Notes

* Frameworks: [link](http://www.softwaretestinghelp.com/test-automation-frameworks-selenium-tutorial-20/),
  + Module based testing framework
  + Library architecture testing framework
  + Data driven testing framework
  + Keyword driven testing framework
  + Hybrid testing framework
  + Behavior driven development framework
* Junit and TestNG
  + TestNG: testing framework inspired from Junit and NUnit. Introduces some new functionalities.
  + TestNG automatically generates a report on the test cases that been run. Just run the tests as TestNG and the reports will be generated
  + Junit will require some setup in order to generate reports
  + TestNG runs all test cases in a specific java package as a suite ([link](http://www.mkyong.com/unittest/testng-tutorial-5-suite-test/)). Requires an xml file that specifies the package to be run.
* Selenium
  + A web testing framework used to mocks button clicks and keys typed on a specific website.
  + Mainly used to execute common user tasks and workflows. Meant to test its functionality.
  + Use Selenium IDE (Firefox plugin) to record/play test cases. Can be exported to JUnit and Java TestNG code.
    - Note: Code generated for JUnit and Java TestNG gets executed too quickly in order to the test case run properly
    - Note: Code generated from selenium IDE for Java TestNG contains depreciated code.
  + Testing workflow idea
    - Capture (button clicks and key presses) using Selenium IDE on Firefox.
    - Play the captured button clicks and/or key presses to ensure the recording was taken properly
    - Export it to JUnit or Java TestNG
    - Add the exported file to testing folder in an eclipse project folder
      * Eclipse project folder will contain the following folders: src, lib, bin, res, doc, and test. The eclipse folder will also contain a few XML files. These XML files are used for compiling source code and running the tests.
      * The exported test case will be stored in the test folder within the eclipse project.
    - Run the ANT script using Command Prompt. The ANT file must be located in the root folder of eclipse project.
      * The ANT file is used for compiling source files, generating a JAR file and executing the test cases
      * Note: Use the following command to compile source files and generate a JAR file:
        + ant –f <location of the ANT file>
      * Note: Run the JAR from the ANT file by using the following commands:
        + Ant –f <location of the ANT file> run
    - All tests are run and a report is generated.

# How to create, manage and run a test case

* Open firefox.
  + Make sure Selenium IDE plugin is installed.
    - Here’s the link for the download. [Link](https://addons.mozilla.org/en-US/firefox/addon/favorites-selenium-ide/)

# Test cases

## Create a production chart

* Create a new workspace
  + Specify a name
  + Provide a description
* Import data using an excel file
* Select Primary attribute: Measure (Capex, Equipment, Fixed Opex, Production, Resources, Variable Opex)
* Go to Dashboard view
* Add a new chart
  + Select chart type: line
  + Select source
    - Select production
* Delete the newly created chart
* Go to Workspaces view
* Delete the newly created workspace