SeleniumJT User guide

SeleniumJT is a Selenium2 wrapper framework that simplifies the coding required for getting you tests up and running fast. In addition to encapsulating the idiosyncrasies of Selenium WebDriver it also supplies a more reliable testing framework for dynamic web pages by allowing elements time for conditions to be met before failure, times that are fully customisable.

# SeleniumJT Technical Specification

* Spring driven.
* Uses Maven.
* Integrated with TestNG
* Supports JUnit & Cucumber
* Utilises the PageObject pattern.

# Installation

1. You will need to install the TestNG Eclipse plugin <http://testng.org/doc/eclipse.html>
2. Clone **ssh://btn-svr-rhl-1/data/gitroot/rdf/Seleniumjt-framework.git**
3. Disconnect from git using **git remote rm origin** or simply remove the .git directory
4. Run ‘mvn clean install -Dmaven.test.skip=true’ (there are no tests for the actual build of this project) and ‘mvn eclipse:eclipse’ to build the project. *There is a convenience ant build that will do this for you* ***selenium-ant/build.xml*** *. You will have to create a my.build.properties file in the ant directory and specify mvn.loc property. There is a sample.my.build.properties file you can copy for this.*
5. Make a copy of sample.my.selenium.properties and call it my.selenium.properties and change the values to suite your environment
6. You will need to configure various Web Browser Drivers. Create a directory and change ‘webdriver.location’ in the my.selenium.properties to point to this directory. Then download the following drivers to that location: *You may need to keep these up-to-date periodically.*
   1. Chrome Driver: <http://chromedriver.storage.googleapis.com/index.html>
   2. IEDriver (32 & 64 bit) <http://selenium-release.storage.googleapis.com/index.html>
   3. PhantomJS <http://phantomjs.org/download.html> (copy the exe here)

**You must run mvn clean install**

# Example Test

The framework comes with a very simple example test. See **com.sample.main.SampleTest.java**.

In order to run this test you will need to

1. Run **com.selenium.tools.CreateTestXml** class and refresh your workspace and navigate to /src/test/xml
   1. Here you will see lots of TestNG files all categorised by browser.
   2. Right-click any of them (for which you have the browser installed) and Run-As -> TestNG suite

# Anatomy of SeleniumJT

Page Object Pattern, the term that selenium users keep buzzing. Page object is a design pattern that can be implemented as a selenium best practices. The functionality classes (PageObjects) in this design represent a logical relationship between the pages of the application.

* The Page Object pattern represents the screens of your web app as a series of objects and encapsulates the features represented by a page.
* It allows us to model the UI in our tests.
* A page object is an object-oriented class that serves as an interface to a page of your AUT

Some of the advantages of page object pattern are:

* Reduces the duplication of code
* Makes tests more readable and robust
* Improves the maintainability of tests, particularly when there is frequent change in the AUT. (Useful in Agile methodology based projects)

[<http://assertselenium.com/automation-design-practices/page-object-pattern/>]

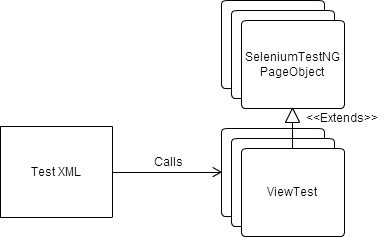
See also <http://martinfowler.com/bliki/PageObject.html>

The use of the PageObject pattern lends itself perfectly to TestNG and SeleniumJT and using this pattern with TestNG makes this framework extremely flexible and powerful.

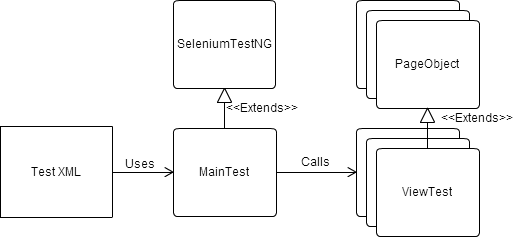
There are three main ways that you can configure and control your tests using PageObjects and TestNG within this framework.

1. Using XML only
   1. Create PageObject tests (classes that extend SeleniumTestNgPageObject) and utilise TestNG’s xml powerful XML functionality to call each PageObjects’ test methods in order whilst making full use of the test data configuration highlighted earlier in this guide.
2. Using XML and Java (as shown in the example earlier)
   1. Create a Java Test class that extends SeleniumTestNG and manage you test data and test dependencies using TestNG’s configurable @Test annotation. Each Test class would call and delegate the testing to one or a set of PageObject tests which in turn extends the PageObject class.

Please refer to <http://testng.org/doc/documentation-main.html>



Using TestNG’s XML to manage and call PageObjects directly



Using TestNG’s XML to a Test class that in turn manages and calls PageObjects

## Examples

The following examples show XML and Java configured TestNG tests. They both perform exactly the same tests in exactly the same order*. NB that to make clear the fact these two approaches do in fact perform exactly the same tests I have refactored both set of tests and introduced a composite class that is shared between both set of tests. This is not a required pattern for SeleniumJT*

### Example 1 by XML

The XML calls the test classes directly. The tests are annotated with @Test.

<suite name=*"Selenium Development Tests"* verbose=*"5"*>

<parameter name=*"browserType"* value=*"chrome"*/>

<test name=*"Sample XML"*>

<classes>

<class name=*"com.rdfgroup.selenium.unit.views.byXml.CoreTestsPageByXml"* >

<methods>

<include name=*"gotoCorePage"*/>

<include name=*"verifyCoreTests"*/>

</methods>

</class>

<class name=*"com.rdfgroup.selenium.unit.views.byXml.AHrefLinksPageByXml"* >

<methods>

<include name=*"gotoLinksPage"*/>

<include name=*"verifyAllLinksWork"*/>

</methods>

</class>

<class name=*"com.rdfgroup.selenium.unit.views.byXml.FormPageByXml"* >

<methods>

<include name=*"goTo"*/>

<include name=*"validateFormElements"*/>

</methods>

</class>

</classes>

</test>

</suite>

**public** **class** CoreTestsPageByXml **extends** SeleniumTestNgPageObject

{

@Test

**public** **void** gotoCorePage() **throws** IOException {

**// tests here**

}

@Test

**public** **void** verifyCoreTests() **throws** IOException {

**// tests here**

}  
}

### Example 2 by Java

The XML calls 1 class, each method is annotated with @Test. The underlying test classes are not marked up with the @Test annotation.

<suite name=*"Selenium Development Tests"* verbose=*"5"*>

<parameter name=*"browserType"* value=*"ff"*/>

<test name=*"This passes"*>

<parameter name=*"title"* value=*"Title is core"*/>

<parameter name=*"properties.file"* value=*"corepage.properties"*/>

<parameter name=*"flow"* value=*"flow1"*/>

<groups>

<run>

<exclude name=*"not\_googlechrome"*/>

</run>

</groups>

<classes>

<class name=*"com.rdfgroup.selenium.unit.tests.AllTests"* >

<methods>

<include name=*"testAllSeleniumCore"*/>

<include name=*"testAllSeleniumLinks"*/>

<include name=*"testAllSeleniumForm"*/>

</methods>

</class>

</classes>

</test>

</suite>

**public** **class** AllTests **extends** SeleniumTestNG

{

@Test

**public** **void** testAllSeleniumCore() **throws** IOException

{

**// NB The PageObject pattern is used here**

CoreTestsPageByJava coreTestsPage = **new** CoreTestsPageByJava(test);

coreTestsPage.gotoCorePage();

coreTestsPage.verifyCoreTests();

}

@Test

**public** **void** testAllSeleniumLinks() **throws** IOException

{

**// NB The PageObject pattern is used here**

AHrefLinksPageByJava aHrefLinksPage = **new** AHrefLinksPageByJava(test);

aHrefLinksPage.gotoLinksPage();

aHrefLinksPage.verifyAllLinksWork();

}

@Test

**public** **void** testAllSeleniumForm(String title) **throws** IOException

{

**// NB The PageObject pattern is used here**

FormPageByJava formPage = **new** FormPageByJava(test);

formPage.goTo();

formPage.validateFormElements();

}

}

**public** **class** CoreTestsPageByJava **extends** PageObject

{

**public** **void** gotoCorePage() **throws** IOException {

**// tests here**

}

**public** **void** verifyCoreTests() **throws** IOException {

**// tests here**

}  
}

# Methods

All of the methods that are available from Selenium Web Driver are also available within this framework. Some of them will be familiar and some methods have been introduced as they perform richer functionality than is provided by WebDriver.

## Verification/Asserting

To verify conditions on your pages this framework uses a straightforward naming convention. Each method starts with ‘verify’ for example verifyTextPresent(text) will see if there is text anywhere on the page and verifyElementText(locator, text) will verify that the element’s value is equal to ‘text’. There are many verify methods for elements (verifyElementXYZ(a, <b>))

Please refer to Appendix A for full JavaDoc.

# Targeting Browsers

Changing which browser to use is simple. At the start of your test suite add the following parameter:

<parameter name="browserType" value="<value>"/>

Where the value can be either

* chrome
* ff
* ie
* safari

Firefox has new releases all the time. This causes problems for WebDriver so it is advisable that you install the latest working Firefox into an alternative director to any that you may be using on a day-to-day basis and make sure it’s auto-updating is turned off. You will need to point SeleniumJT to this installation directory by setting the property ‘firefox.exe.location’ in your my.selenium.properties.

# Properties and Test Data

All your properties are located in src/main/resources/properties. Properties that are global should be set in selenium.properties. Properties that would be different from environment to environment such as multiple developer machines, dev, uat and staging should be placed in my.selenium.properties (which should not be version controlled).

## selenium.properties

This is the Selenium runtime properties that allow you to configure the behaviour of your tests, how many times an element should be tested for a condition before it fails and the period of time between checks etc.

## my.selenium.properties

This file is optional and should not be committed to any version control. It allows localisation of custom preferences to be used when running tests such as the installation location of Firefox. Any property named here will overwrite same named properties in the selenium.properties file.

## testdata.properties

This is the global test data properties. Here you can put in data that is useful throughout all your tests. Data added to this file will be available within your code using test.data(“key”)

**Example**  
mykey=This is some test data

## TestNG Parameters

Leveraging TestNG’s parameter paradigm you are able to add test data within your text XML. This can be at the suite level or at the test level

**Example**  
<parameter name="key" value="Some Value"/>

Data added this way will also be available within your code using test.data(“key”). It will also be available to your TestNG tests in the usual way via the standard @Parameter annotation.

## Custom Property files

Using TestNG’s parameter paradigm you are able to declare a custom properties file to use for your test or test suite

**Example**  
<parameter name="properties.file" value="mycustom.properties"/>

The name must be “properties.file” and the value must match the name of a properties file that you must place in src/main/resource/properties directory. You may place this in a subdirectory as long as your parameter value reflects this i.e. “subdirectory/myfile.properties”. Data added this way will also be available within your code using test.data(“key”)

# Default properties

## debugMode

By Utilising Aspect Orientated Programming it is possible for you to turn on a verbose debug mode that will output each and every call within the SeleniumJT core classes. This is useful when you wish to watch what is happening with each test that you create. To enable this mode simple change ‘debugmode’ within selenium.properties to ‘true’.

## WAITING\_TIME

This property is set in seconds. The time you should wait for any condition to fail. This forms the basis for the condition ‘waitUntil…’.

## RETRY\_ATTEMPTS & COMMAND\_REPEAT\_MILS

These two properties combine to provide a timing for verifying elements such as ‘verifyElementContains’, verifyElementPresent’ and verifyTextPresent’. It allows for pauses caused by elements dynamically loaded via Ajax or JavaScript, conditions that with the standard Selenium framework would fail immediately if the conditions are not met at the time of execution.

target.website

This is simply the URL of the website that is under test. It prevents the requirement to type the full URL in any open(String url) method call so instead of typing test.open(“http://someurl.com:8080/someUri”) you can type test.open(“/someUri”). You could however not specify this value and instead type the full URL in each open() method call but this is not advisable due to the possibility of testing against different environments such as DEV, STAGING and UAT. NB that when you deploy your selenium tests across different environments you can specify a different target website in my.selenium.properties for each environment. This will overwrite any property set in selenium.properties.

## screenshot.loc

You can perform screenshots at any time. Each screenshot must have a filename which you supply to the screenshot method. The filename must be a ‘\*.gif’. The screenshot.loc property determines where these gifs are saved to.

## ieDriverVersion

There are two ieDrivers to choose from, 32 and 64bit. If this property is not set then the 32bit version is used by default. If you want to use the 64bit version then set ieDriverVersion=64.

# Building Text XML using Ant

There exists some convenience targets defined within ant/tests-build.xml scripts. The most convenient being ‘create-testng-xml’ that will create dummy testing xml files for every browser for you and place the results within target/xml. These will be fully runnable tests and serve as a great start to configuring testing for your new test classes. This script will look at all classes that reside within a particular package root and is configurable using the testing.xml.build.properties file.

NB: There are only 2 important properties within testing.xml.build.properties

* test.package. The package root of all tests that you want XML for.
* target.file.dif. where the xml is to be saved.

# Converting from Selenium IDE

Using the Firefox Selenium IDE plugin is a convenient way to record your tests. You can export your tests as TestNG or JUnit Java code but this code will not conform to SeleniumJT. To help the transition from IDE to this framework there is a simple ant task that will convert the code to use with this framework. The conversion is not 100% but it will greatly reduce any typing that would otherwise be required. To use this feature you will use Ant but before you do anything you will need to set your maven repository location in testing.xml.build.properties so Ant can locate the appropriate classes for the conversion.

1. Once you used Selenium IDE and are ready to export your test, export it to /src/main/ide/ide.in.txt from within your test project.
2. Run the Class **com.selenium.tools.ConvertIDE**
3. The result will be printed to the console
4. You can then copy the code to the appropriate test class.

The conversion tool uses a pattern matching system that can be expanded. There is a file /resource/ideconvertor/selenium-matchers.properties file that allows you to write your own pattern matching.

**Example**  
assertEquals(selenium.getText("<$>"), "<$>");=test.verifyElementText("<$>", "<$>");

The text to the left of the ‘=’ is what to search for and the text on the right is what to replace it with. The ‘<$>’ is there to ensure that the custom values form the left are transferred verbatim to the replacement text.

So assertEquals(selenium.getText("someId"), "someValue");

Would become test.verifyElementText("someId", "someValue");

# Core development

You do not need to read this section unless you are making changes to the SeleniumJT core project. The SeleniumJT framework has a maven dependency for this project and you should not need to do anything here.

## Installation

The Core development comes in 3 parts. The core code, a test server and unit tests. If you are to do any work on the core then you need to download all 3 projects as you will need to run the unit tests against the test server to ensure you haven’t broken anything.

**Clone ssh://btn-svr-rhl-1/data/gitroot/rdf/Seleniumjt-core.git**

In here you will have

### core-framework

Import this as a project into eclipse. This is the main core framework that is used to build and deploy to a maven repository.

### core-framework-tests/server

Import this as a project into eclipse. This is a stand-alone application that runs on a jetty-server based on The Stack project. You will need to add a key/value in your eclipse’s **String Substitution**. Variable=M2\_HOME, value={Path to your maven directory}. This server provides pages and forms that the unit-tests will test against. To start the server you run ‘Jetty start.launch’

### core-framework-tests/unit-tests

Import this as a project into eclipse. After every change to the **core-framework** you must start the test server (above) and run the unit tests found under **src/main/xml**. Right click these and **Run As \_> TestNG Suite**

You will need TestNG installed within eclipse.

You should have a green test run. Any changes you make to the core make sure these tests run green before you commit any changes.

# Appendix A – JavaDoc

## Class SeleniumJT

java.lang.Object

extended by **com.jt.selenium.SeleniumJT**

@Component

public class **SeleniumJT**

extends java.lang.Object

|  |  |
| --- | --- |
| **Field Summary** | |
| java.util.Properties | [**testData**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#testData) |

|  |  |
| --- | --- |
| **Constructor Summary** | |
| [**SeleniumJT**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#SeleniumJT())() |  |

|  |  |
| --- | --- |
| **Method Summary** | |
| void | [**attachFile**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#attachFile(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String fileUrl) Selenium's attachFile does not work - so I've written my own |
| boolean | [**attachmentCompatibleBrowser**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#attachmentCompatibleBrowser())() |
| void | [**blur**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#blur(java.lang.String))(java.lang.String locator) |
| void | [**causeFailure**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#causeFailure(java.lang.String))(java.lang.String message) |
| void | [**check**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#check(java.lang.String))(java.lang.String locator) |
| void | [**checkServiceAvailability**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#checkServiceAvailability())() Make sure site is up and not being updated |
| void | [**clearLocalStorage**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#clearLocalStorage())() |
| void | [**click**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#click(java.lang.String))(java.lang.String locator) |
| void | [**click**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#click(java.lang.String, int))(java.lang.String locator, int pause) |
| void | [**clickAndWait**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#clickAndWait(java.lang.String))(java.lang.String locator) |
| void | [**clickAndWaitForAjax**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#clickAndWaitForAjax(java.lang.String))(java.lang.String locator) |
| void | [**clickNoValidation**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#clickNoValidation(java.lang.String))(java.lang.String locator) |
| void | [**closeAlert**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#closeAlert())() |
| void | [**closeBrowser**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#closeBrowser())() |
| void | [**copy**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#copy(java.io.File, java.io.File))(java.io.File src, java.io.File dst) |
| java.lang.String | [**data**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#data(java.lang.String))(java.lang.String key) |
| void | [**debugPause**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#debugPause(int, java.lang.String))(int seconds, java.lang.String message) |
| void | [**dragAndDropToObject**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#dragAndDropToObject(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String targetLocator) |
| void | [**dragAndDropWithPause**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#dragAndDropWithPause(java.lang.String, java.lang.String, int))(java.lang.String locator, java.lang.String targetLocator, int pause) |
| void | [**executeJavaScript**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#executeJavaScript(java.lang.String))(java.lang.String script) |
| void | [**fireEvent**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#fireEvent(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String eventName) This requires a jquery locator or an ID. |
| void | [**focus**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#focus(java.lang.String))(java.lang.String locator) |
| java.lang.String | [**getAttribute**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#getAttribute(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String attribute) |
| com.jt.selenium.configuration.SeleniumConfiguration | [**getConfiguration**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#getConfiguration())() |
| org.openqa.selenium.WebElement | [**getElement**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#getElement(java.lang.String))(java.lang.String locator) |
| org.openqa.selenium.WebElement | [**getElementByTagName**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#getElementByTagName(java.lang.String))(java.lang.String locator) |
| java.lang.String | [**getHtmlSource**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#getHtmlSource())() |
| java.lang.String | [**getInputValue**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#getInputValue(java.lang.String))(java.lang.String locator) |
| java.lang.String | [**getItemFromLocalStorage**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#getItemFromLocalStorage(java.lang.String))(java.lang.String key) |
| java.lang.String | [**getKeyFromLocalStorage**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#getKeyFromLocalStorage(int))(int key) |
| java.lang.Long | [**getLocalStorageLength**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#getLocalStorageLength())() |
| java.lang.String | [**getLocation**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#getLocation())() |
| int | [**getOccurrences**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#getOccurrences(java.lang.String))(java.lang.String regex) |
| java.lang.String | [**getText**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#getText(java.lang.String))(java.lang.String locator) |
| java.lang.String | [**getValue**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#getValue(java.lang.String))(java.lang.String locator) |
|  |  |
| boolean | [**isElementPresent**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#isElementPresent(java.lang.String))(java.lang.String locator) |
| boolean | [**isItemPresentInLocalStorage**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#isItemPresentInLocalStorage(java.lang.String))(java.lang.String item) |
| boolean | [**isTextPresent**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#isTextPresent(java.lang.String))(java.lang.String text) |
| boolean | [**isVisible**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#isVisible(java.lang.String))(java.lang.String locator) |
| void | [**open**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#open(java.lang.String))(java.lang.String url) |
| void | [**pause**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#pause(int))(int millisecs) |
| void | [**removeItemFromLocalStorage**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#removeItemFromLocalStorage(java.lang.String))(java.lang.String item) |
| void | [**restartBrowser**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#restartBrowser())() |
| void | [**returnToFrameParent**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#returnToFrameParent())() To be used in conjunction with selectFrame() to return to the parent frame. |
| void | [**runJavaScript**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#runJavaScript(java.lang.String))(java.lang.String script) |
| boolean | [**runningInChrome**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#runningInChrome())() |
| boolean | [**runningInFirefox**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#runningInFirefox())() |
| boolean | [**runningInIE**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#runningInIE())() |
| boolean | [**runningInSafari**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#runningInSafari())() |
| void | [**select**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#select(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String value) |
| void | [**selectAndWait**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#selectAndWait(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String value) |
| void | [**selectFrame**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#selectFrame(java.lang.String))(java.lang.String targetFrame) |
| void | [**selectWindow**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#selectWindow(java.lang.String))(java.lang.String locator) |
| void | [**setConfiguration**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#setConfiguration(com.jt.selenium.configuration.SeleniumConfiguration))(com.jt.selenium.configuration.SeleniumConfiguration configuration) |
| void | [**setItemInLocalStorage**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#setItemInLocalStorage(java.lang.String, java.lang.String))(java.lang.String item, java.lang.String value) |
|  |  |
|  |  |
| void | [**stop**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#stop())() |
| void | [**type**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#type(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String value) |
| void | [**typeHidden**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#typeHidden(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String value) |
| void | [**typeTinyMceEditor**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#typeTinyMceEditor(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String value) Custom method for typing text into a tinyMce which is formed of an embedded iframe html page which we need to target and type into. |
| void | [**uncheck**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#uncheck(java.lang.String))(java.lang.String locator) |
| void | [**validatePage**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#validatePage(java.lang.String))(java.lang.String identifier) |
| void | [**verifyAlert**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyAlert(java.lang.String))(java.lang.String regex) |
| void | [**verifyAttribute**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyAttribute(java.lang.String, java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String attribute, java.lang.String value) |
| void | [**verifyChecked**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyChecked(java.lang.String))(java.lang.String locator) |
| void | [**verifyCssClass**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyCssClass(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String value) |
| void | [**verifyElementContains**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyElementContains(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String value) Looks for a substring within the element |
| void | [**verifyElementNotContains**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyElementNotContains(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String value) |
| void | [**verifyElementNotPresent**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyElementNotPresent(java.lang.String))(java.lang.String locator) |
| void | [**verifyElementNotVisible**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyElementNotVisible(java.lang.String))(java.lang.String locator) |
| void | [**verifyElementPresent**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyElementPresent(java.lang.String))(java.lang.String locator) |
| void | [**verifyElementText**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyElementText(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String value) Verifies the value of text within the given element. |
| void | [**verifyElementText**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyElementText(java.lang.String, java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String value, java.lang.String message) Allow custom message to verifyElementText |
| void | [**verifyElementTextNotEqualTo**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyElementTextNotEqualTo(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String value) |
| void | [**verifyElementVisible**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyElementVisible(java.lang.String))(java.lang.String locator) |
| void | [**verifyImageSource**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyImageSource(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String src) Verify that the src of the image [identified by it's id] is correct |
| void | [**verifyInputValue**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyInputValue(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String value) |
| void | [**verifyInputValue**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyInputValue(java.lang.String, java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String value, java.lang.String msg) |
| void | [**verifyInputValueTinyMce**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyInputValueTinyMce(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String value) |
| void | [**verifyLocalStorageValue**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyLocalStorageValue(java.lang.String, java.lang.String))(java.lang.String key, java.lang.String value) |
| void | [**verifyNoJavaScriptErrorOnPage**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyNoJavaScriptErrorOnPage())() Make sure no JavaScript error occurred. |
| void | [**verifyNotChecked**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyNotChecked(java.lang.String))(java.lang.String locator) |
| void | [**verifyOccurrencesInElement**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyOccurrencesInElement(java.lang.String, java.lang.String, int))(java.lang.String locator, java.lang.String regex, int occurs) |
| void | [**verifySelectedLabel**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifySelectedLabel(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String value) |
| void | [**verifySelectedLabelContains**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifySelectedLabelContains(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String value) |
| void | [**verifyTextNotPresent**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyTextNotPresent(java.lang.String))(java.lang.String text) |
| void | [**verifyTextNotPresent**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyTextNotPresent(java.lang.String, java.lang.String))(java.lang.String text, java.lang.String message) |
| void | [**verifyTextPresent**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyTextPresent(java.lang.String))(java.lang.String text) This text should appear on the screen. |
| void | [**verifyTextPresent**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyTextPresent(java.lang.String, java.lang.String))(java.lang.String text, java.lang.String msg) This text should appear on the screen |
| void | [**verifyTitle**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#verifyTitle(java.lang.String))(java.lang.String title) |
| void | [**waitForAlert**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#waitForAlert())() |
| void | [**waitForElementNotPresent**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#waitForElementNotPresent(java.lang.String))(java.lang.String locator) |
| void | [**waitForElementNotVisible**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#waitForElementNotVisible(java.lang.String))(java.lang.String locator) |
| void | [**waitForElementPresent**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#waitForElementPresent(java.lang.String))(java.lang.String locator) |
| void | [**waitForElementPresentAndVisible**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#waitForElementPresentAndVisible(java.lang.String))(java.lang.String locator) |
| void | [**waitForElementText**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#waitForElementText(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String text) |
| void | [**waitForElementVisible**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#waitForElementVisible(java.lang.String))(java.lang.String locator) |
| void | [**waitForText**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#waitForText(java.lang.String))(java.lang.String text) After clicking something that invokes an Ajax event we need to wait for some value to return before we can continue. |
| void | [**waitForTitle**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#waitForTitle(java.lang.String))(java.lang.String title) |
| void | [**waitForTitleToContain**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#waitForTitleToContain(java.lang.String))(java.lang.String text) |
| void | [**waitForValueInElement**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#waitForValueInElement(java.lang.String, java.lang.String))(java.lang.String locator, java.lang.String text) |
| void | [**windowFocus**](file:///D:\Workspace\Selenium\Selenium-JT-WebDriver-Core\doc\com\jt\selenium\SeleniumJT.html#windowFocus())() |