5 copy

~~Faculty~~
HARISHANKAR.R

Selenium

Features:

1. Used to check only websites
2. It is very flexible. It works with any browser, any Operating System or any programming language.
3. Used for Functionality and Regression testing. It can't be used for performance testing
4. It was developed by Thought works to test the functionality of websites
5. Supports all programming languages(HTML,C, C#, Perl, Python, Ruby, Java etc)
6. It has 3 components (Selenium IDE, Selenium RC and Web Driver) ie. Selenium 2.0
7. Selenium Grid is under development.
8. Selenium Core is a part of Selenium IDE, Selenium RC and Web Driver. Selenium Core contains a set of Java Script library functions
9. Selenium IDE is the interface for recording, running, interpreting, debugging as well as play back the script.
10. Selenium IDE is a Firefox plug-in, so it can be used only with a Firefox browser.
11. To play scripts in other browsers we can use Selenium RC.
12. Official website is http: /selenium hq.org/
13. Can be used to test website developed in Web 2.0 { Microsoft introduced Silver Light/AJAX/XAML – Web 2.0 Technologies}

Selenium 2.0 → Advanced with many commands (new 2.0 websites)
Selenium 1.0 → Limited features and commands (old websites)

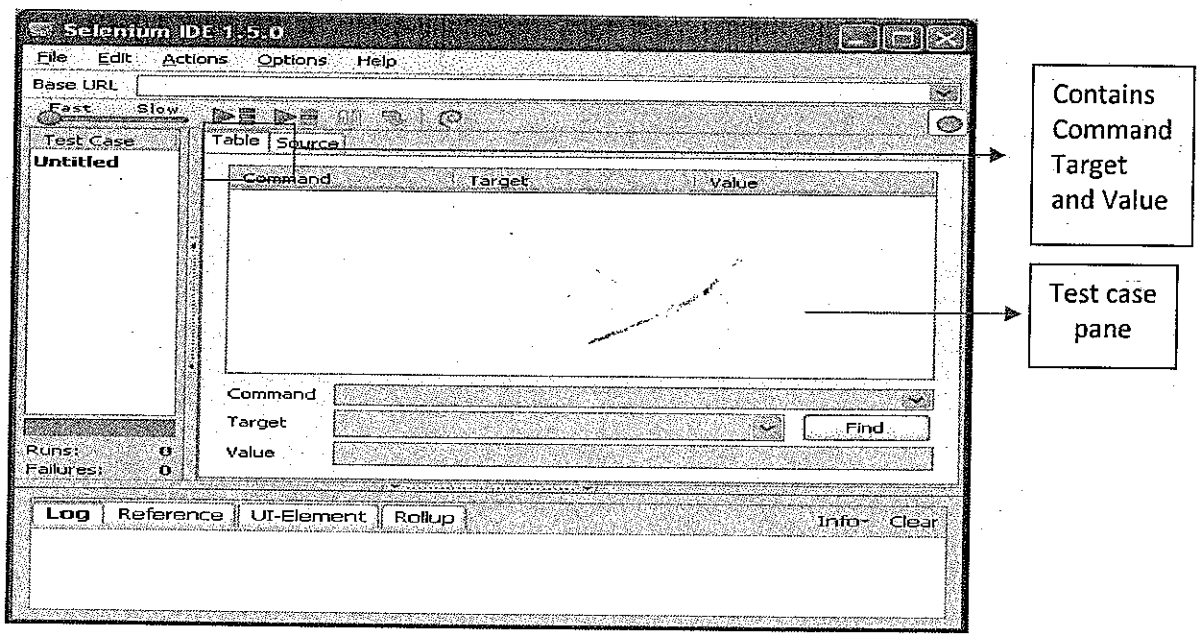
Selenium Core:

- Core Engine of Selenium
- Contains Java Script/DHTML Libray
- It's now incorporated into all the components of Selenium like IDE and RC
- Supports a variety of platforms:
  - Windows IE 6.0 and 7.0 , Firefox 0.8 to 2.0
  - Mac OS X: Safari2.0.4+,Firefox 0.8 to 2.0
  - Linux: Firefox 0.8 to 2.0

Installing Selenium IDE as a Firefox Plug-in:

1. Prerequisites: Ensure JDK Kit is installed (Java™ SE Development Kit 6 Update 2.7) (Check in Control Panel Add/Remove Programs)
2. Install Mozilla Firefox browser
3. Click on http://seleniumhq.org/download.
4. Click Download, Select the latest version, click on Firefox extensions link.
5. When the Software Installation Window appears select "Selenium IDE"
6. Press Install button
7. "Selenium IDE" Add on is installed
8. Click on "Restart Firefox" button
9. Open Firefox browser go to Tools and verify whether "Selenium IDE" is displayed.
10. If found, then Selenium IDE installation is complete.
11. Selenium IDE can be opened from Firefox either by
    View →SideBar-→Selenium IDE(Records with the same window)
    Tools → Selenium IDE (Records the functional flow even if it's in a new window)

Now Open Selenium IDE from Tools Menu. It opens with an empty script editing window



Test case pane: In this pane the Script is displayed in 2 tabs Table and Source.
The table pane contains Command, Target and Value entry fields which displays the currently selected command along with its parameters. These are entry fields where you can modify the currently selected command.
The Source pane displays the test case in the native format in which the file will be stored
It contains 4 tabs ie. Log, Reference, UI Element, Rollup
Log: When you run your test case, error messages and information messages showing the progress are displayed in this pane automatically.
Reference: Displays documentation on the current command
Rollup: The single command is referred to simply as a "rollup".
UI Element Locator: A mapping between meaningful names for a page element.
To Record a Script: Open Selenium IDE. In the URL type : www.book.theautomatedtester.co.uk. (The sample application of Selenium IDE). Ensure that the browser is also open at the same webpage.

Selenium is in a default recording mode. Do some recording activities on the sample application. The Selenese script is generated in the Table pane and as a HTML script in the Source pane.

To save the script

Select File in the Menu Bar and Select Save Test Case. Give a test case name example: Test1.html

To save a test suite:

Create 2 test cases and save it as Test1.html and Test2.html
To save these test cases as a Suite file Select File → Save Test Suite As
Now if you open the Suite file you can find the links to the 2 testcases test1 and test2.
To execute a Test Suite click on play Test Suite icon.

To execute a script from a specific step:

Select the specific step. Select Actions -→ Set/Clear Start Point. A green icon will be displayed in the Table. Press **playtestcase** icon to run a single test case from the Start Point.

```
<html>
 <body>
  <div id="pancakes">
   <button type="button" name="pancake" value="Blueberry">Blueberry</button>
   <button type="button" name="pancake" value="Banana">Banana</button>
   <button type="button" name="pancake" value="Strawberry">Strawberry</button>
  </div>
 </body>
</html>
```

Scenario:

We just added a strawberry pancake in our application and we want to test that the button that adds it into the cart works. With a locator like name=pancake, Selenium will find 3 elements and return the first one: the test will never fail even if the strawberry button is not here! Use a value filter like name=pancake value=Strawberry and the locator successfully identifies the Strawberry button.

Index : Same as name but works with an index. Using the previous example, the locator name=pancake index=2 will select the Strawberry button.

Note: Index starts at 0

3. Location by id

Id=id

Selects the element with @id attribute

The Id strategy looks for an element in the page having an `id` attribute corresponding to the specified pattern. `<label id="my_id" />` will be matched by a locator like `id=my_id` or just `my_id`

4.Locating by DOM

The Document object model represents an Html document and can be accessed using Java Script. It works by locating elements that matches the java script expression referring to an element in the DOM of the page.

Dom=javascriptExperssion(Refer earlier example)

dom=document.div['pancakes'].button[0]

document.div[0].button[2]

dom=function foo() { return document.getElementById("pancakes"); }; foo();

5.Locating by XPath

xpath=xpathExpression

Locates an element with XPath Expression

XPath is the language used for locating nodes in an XML document. As HTML can be an implementation of XML(XHTML), Selenium users can leverage this powerful language to target elements in their web applications.

One of the main reasons for using XPath is when you don't have a suitable id or name attribute for the element you wish to located. You can use XPath to either locate the element in abosolute terms or relative to an element that does have an id or name attribute.

By using Relatlve XPath we can find a nearby element with an id or name attribute (ideally a parent element), we can locate the target element based on the relationship. This is less likely to change and can make your tests more robust.

Since only xpath locators start with "//", it is not necessary to include the xpath=label when specifiying an XPath locator

Example

xpath=//img[@alt='The image alt text']

xpath=//table[@id='table1']//tr[4]/td[2]
xpath=//a[contains(@href,'#id1')]
[Refer earlier example for the following]
xpath=//button[@value="Blueberry"]: matches the Blueberry button
//div[@id="pancakes"]/button[0]: does the same thing

When the element Id is changing continuously we can use XPath in its location to identify the element. For example as Google is changing continuously we can use XPath expression.

6. Locating Hyperlinks by link Text

Link=textPattern

Selects a link with the specified text. If two links with the same text are present, then the first match will be used

Eg: In an HTML if any text is comes under the anchor '<a>' tag then we use the link= textPattern

Link Text
link=Link Text

7. Locating by CSS

CSS(Cascading Style Sheets) is a language for describing the rendering of HTML and XML documents. CSS uses Selectors for binding style properties to elements in the document. These selectors can be used by Selenium as another locating strategy

css=cssSelectorSyntax
css=div[id="pancakes"] > button[value="Blueberry"] selects the button with its value property set at Blueberry if children of the pancakes div

Matching text patterns

Like locators, patterns are a type of parameter frequently required by Selenese commands. Examples of commands which require patterns are verifyTextPresent, verifyTitle, verifyAlert, assertConfirmation, verifyText, and verifyPrompt.

Patterns allow you to describe, via the use of special characters, what text is expected rather than having to specify that text exactly.

Globing, Regular Expression and Exact are the 3 method used for matching text patterns

Globbing: Globs are used extensively throughout Selenium as the default pattern matching technique. Most users are normally looking for part of a sentence when verifying or asserting in their tests.

An * translates to "match anything," i.e., nothing, a single character, or many characters.

[ ] (character class) which translates to "match any single character found inside the square brackets."

A dash (hyphen) can be used as shorthand to specify a range of characters (which are contiguous in the ASCII character set).

Eg:

[aeiou] matches any lowercase vowel

- [0-9] matches any digit
- [a-zA-Z0-9] matches any alphanumeric character

Example:

Record for a script. Click on the link Chapter 3. Select the word 'cool" right click and select verifyTextPresent.

Record for some more activities. Execute the script. It fails as the value of 'cool' changes during playback

Now introduce globing by adding glob:*ool in the Target field. Execute the script and now view it.

The script executes without any errors.

RegularExpressions:

Selenese regular expression patterns offer the same wide array of special characters that exist in JavaScript

.	**Any single character**
[ ]	**Character class:any single character that appears inside the brackets**
*	**Quantifier: 0 or more of the preceding character(or group)**
+	**Quantifier: 1 or more of the preceding character (or group)**
?	**Quantifier:0 or 1 of the preceding character**
{1,5}	**Quantifier: 1 through 5 of the preceding character**
|	**Alternation: the character/group on the left or the character/group on the right**
( )	**Grouping: Often used with alternation and / or quantifier**

Record for yahoo.com. Select the date Thu, Jul 19,2012 .Right click and select verifyTextPresent Thu, Jul 19,2012.

If the script is executed will pass but will fail if executed on the next day

By using regular expression ie. Regexp: Thu, Jul [0-9]{1,2}, 2012 the script can be executed.

Exact:

It does not use any special characters .

If you needed to look for an actual asterisk character (which is special for both globbing and regular expression patterns), the exact pattern would be one way to do that.

Eg: To check for the entire text with the character : in the title, Selenium: Beginners Guide we can use the exact pattern.In the above example we use exact: Selenium: Beginners Guide

Echo: It's a selenese command to print the result of a command or some other information to the log window during execution

Eg: In the above example we use a storeElement Command to store the value of the button "Verify this button is here".

The storeElement command stores the value of the button in a variable say "a". Now during exeution it checks if the button value is the same. It returns TRUE if the button value is the same and viceversa. Now the value of variable a is displayed in the log window using an echo command with target as ${a}