Selenium IDE and Locators

Presenter:
Manu Parashar
12/24/2013
Mindfire Solutions

About Me:

Manu Parashar

Skills:

Automation: Selenium Webdriver, Selenium RC, QTP 11

Frameworks: Junit, Testng

PL: Java, C#

Connect Me:

LinkedIn: http://www.linkedin.com/in/parasharmanu

Google+: https://plus.google.com/110738751918785957229/

Contact Me:

Email: manup@mindfiresolutions.com/ manu.aug5@gmail.com

Skype: mfsi_manup

Agenda

1. Selenium IDE
   - Introduction to the IDE
   - IDE Installation
   - Features of IDE
   - Creating Test cases using Record feature
   - Executing Test cases using Play feature

2. Locators
   - Types of Locators
   - Locating Techniques
   - Locators examples

Recap

- What is Selenium?
- Features of Selenium?
- Tools under Selenium's Tool Suite?
- How Remote Control(RC) works?
- How Webdriver works?
- Features of Selenium IDE?
- Different types of Locators?

Introduction to Selenium IDE

Integrated development environment for Selenium scripts.

Contains a context menu that allows to first select a UI element from the browser's current page and then select from a list of Selenium commands with parameters pre-defined according to the context of the selected UI element.

Includes the entire Selenium Core, allowing you to easily and quickly record and play back tests in the actual environment that they will run in.

It is not only a recording tool: it is a complete IDE. Either its recording capability can be used, or scripts may be edited.

IDE Installation

Selenium IDE is distributed as Firefox extension:

- Go to the Selenium website (http://www.seleniumhq.org) or Firefox add-ons website (http://addons.mozilla.org).
- Download the latest release.
- Follow the instructions and restart Firefox.
- Selenium IDE is accessible from the 'Tools' Menu of Firefox .

Features of IDE

- Easy-to-use Firefox plug-in
- Easy record and playback
- Intelligent field selection will use IDs, names, or XPath as needed
- Autocomplete for all common Selenium commands
- Debugging feature with step-by-step and breakpoints
- Save tests as HTML, Ruby scripts, or any other format
- Option to automatically assert the title of every page
- Easy customization through plugins

Creating Test Cases

Recording

During recording, Selenium-IDE will automatically insert commands into your test case based on your actions.

Record button is ON by default

Adding Verifications and Asserts With the Context Menu

Editing

Insert Command

Insert Comment

Edit a Command or Comment

Executing Test Cases

- Run a Test Case (clicking Run button)
- Run a Test Suite (clicking Run All button)
- Stop and Start (Pause/Resume button)
- Stop in the Middle (Toggle Breakpoint option)
- Start from the Middle (Set/Clear Start Point option)
- Run Any Single Command (Double click)

Selenese

Domain Specific Language(DSL) to write tests. Simple xHTML tables of 3 columns and 'n' rows:

- A method on 1st column: the action to perform
- A locator on 2nd column: the subject of the method (ie: what element is acted upon)
- A value on 3rd column: the value to be passed to the method (what to type in a text field for instance)

Selenium commands that can be categorized as:

- Actions: commands that manipulate the state of application like clicking links or buttons.
- Accessors: examine the state of the application and store values in variables.
- Assertions: are like Accessors, but they verify that the state of the application conforms to what is expected. Eg. "assert", "verify", and " waitFor"

Locators

Attributes-based locators:

It relies on the attributes of the elements to locate them.

- Identifier
- Id
- Name
- Link
- CSS

Structure-based locators:

It rely on the structure of the page to find elements.

- DOM
- XPath
- CSS

Locating Techniques

```
<html>
 <body>
  <form id="loginForm">
   <input name="username" type="text" />
   <input name="password" type="password" />
   <input name="continue" type="submit" value="Login" />
   <input name="continue" type="button" value="Clear" />
   <p>Are you sure you want to do this?</p>
   <a href="continue.html">Continue</a>
   <a href="cancel.html">Cancel</a>
  </form>
</body>
<html>
```

Identifier: works with the id and name attributes of html tags.

Eg. identifier=loginForm, identifier=password

Id: more limited than the identifier locator, but more explicit.

Eg. id=loginForm

Name: locate the first element with a matching name attribute.

Eg. name=username, name=continue value=Clear

Link: locates a hyperlink by using the text of the link

Eg. link=Continue, link=Cancel

CSS: Selectors for binding style properties to elements in the document

Eg. css=form#loginForm, css=input[name="username"],
css=input.required[type="text"], css=input.passfield,

DOM: works by locating elements that matches the javascript expression refering to an element in the DOM of the page.

Eg. dom=document.getElementById('loginForm'),
dom=document.forms['loginForm'], dom=document.forms[0],
document.forms[0].username, document.forms[0].elements[3]

Xpath: language used for locating nodes in an XML document.

Eg. xpath=/html/body/form[1] – *Absolute path (would break if HTML was changed only slightly)*

//form[1] (3) – *First form element in the HTML*

xpath=//form[@id='loginForm'] (3) – *The form element with attribute named 'id' and the value 'loginForm'*

//input[@name='username'] (4) – *First input element with attribute named 'name' and the value 'username'*

//form[@id='loginForm']/input[1] (4) – *First input child element of the form element with attribute named 'id' and the value 'loginForm'*

Questions
??

Thank you

Presenter: Manu Parashar, Mindfire Solutions