Web Application Testing With Selenium IDE

Regina Ranstrom
Overview

- When do automate testing and when is manual testing sufficient?
- Using an automated testing tool, how can one ensure sustainability?
- What features would make Selenium more usable? What are some setbacks?
- Why is Selenium suitable for testing an application like the vEOC?
Automated vs. Manual Tests

- Automation ensures that the software is being tested thoroughly and often.
- Manual testing is tedious and inefficient.
- When modifications are made, Automation ensures that the same checks are being made each time.
- Scripting conveniences, such as loops and data storage, come in handy.
A Brief History

- Before Selenium, there was no way to write automated tests for web sites that used JavaScript heavily.
- Old technique: write server-side code in place of JavaScript, test using jWebUnit (which was inefficient)
- There were JavaScript unit test libraries, but no functional/UI testing tools
- No tool that could drive multiple browsers
- Selenium is got a lot of attention with the rise of Ajax, Rails, and Web 2.0
Selenese

- Selenese is the set of commands to test your web software in the following ways:
  - support testing of window size, mouse position, alerts, Ajax functionality, pop up windows, event handling, and other web-application features
- Pros and Cons: Data Storage --> one variable
- Dynamic Pages require user extension (goto if)
Actions, Accessors, and Assertions

- Actions manipulate the state of the application
  - (i.e. “click this link,” “select that option”)
- Accessors examine the state of the application and store the results in variables
- Assertions verify that the state of the application conforms to what is expected.
- All Selenese commands are based on these three things
Predicting Commands (Pros and Cons)

- Right clicking anywhere on the page will prompt a verification of a command for that content.
- Selenium-IDE will attempt to predict what command, along with the parameters, you will need for a selected UI element on the current web-page.
- Not all elements are viewable, like window ID’s or the XPath of an element in a drop down menu.
- DOM inspector makes this job much easier, this interface viewing tool could easily be added to Selenium.
Capturing actions is not perfect
Selenium makes it easy to edit parts of the script that are incorrect
To edit, you need to have an understanding of the DOM
Capturing is done after the program is written
However, Selenium tests can also be written before and imported
“waitFor” commands are a type of assertion that waits for some condition to become true (which can be very useful for testing Ajax applications).

Clicking a link in an Ajax app. does not signal the entire page to reload

Because of this, it is often necessary to wait for certain elements

One slow response or even an error will not get in the way of other tests being run
Sustainable Tests

- It is ideal to create tests that are specific to a small part of the program
- They are easily modifiable and re-usable
- Tests in Selenium cannot be embedded in one another, only executed sequentially
- Because of this it is sometimes hard to create and save specific test cases
- Choosing locators based on their ID’s make tests very adaptable
An Example Test
Limitations

- **Browser**
  - Selenium IDE (by itself) can only be used in Firefox.

- **Language**
  - Selenium can only execute scripts created in Selenese

- It is difficult to use Selenium for checking complex test cases involving dynamic components (without a user extension)

- The important part of testing is the results, however, Selenium’s log is not exportable!
References

- http://seleniumhq.org/docs/03_selenium_ide.html#id4 http://www.infoq.com/articles/testing-ajax-selenium
- “An Introduction to Web Applications with twill and Selenium,” C. Titus Brown, Grig Gheorghiu, and Jason R. Huggins
- “When Should a Test be Automated?” Brian Marick
- “Automating Functional Tests Using Selenium” Antawan Holmes and Marc Kellogg
- “A Look at Selenium” Grig Gheorghiu
- “Improving Web Application Testing with User Session Data” Sebastian Elbaum, Srikanth Karre, and Gregg Rotherme
- http://www.w3.org/DOM/#what
- “Refactoring Test Code” Arie van Deursen, Leon Moonen, Alex van den Bergh, Gerard Kok
- “Project ENSAYO: A Virtual Emergency Operations Center for Disaster Management Research, Training and Discovery” Irma Becerra-Fernandez, Michael Prietula, Greg Madey, Domingo Rodriguez
- http://httpunit.sourceforge.net/