Extension of Selenium db for Better Compatibility with the Database for Web Based Application Testing

Rupandeep Virk and Neha Malhotra

Abstract

Number of softwares have been created as web based applications (applications that are executed in a web browser) now a days. Quality is the main aspect that needs to be assured before the softwares are delivered to the customers. Many testing tools are available in market for testing of these softwares. Selenium is one such tool available. Selenium RC tool, one of the components of Selenium available is used to create automated test cases for web based applications. This paper gives an introduction of selenium and its components. Selenium RC one of the component of Selenium, basically used for web application automation testing has some assert functions that help query the database for real time data. In this paper I am going to propose a novel approach in selenium RC tool. New assert functions are to be added into the tool to extend SeleniumDB management systems, such as Oracle or SQL Server. New assert functions are to be added to help provide different tests in a database including the assert functions that, for an instance can be used to perform a join among different tables in a database.

Keywords: Testing tools, Selenium RC, Asserts.

1. Introduction

Software testing is a process which is used to identify error or faults in a system to make it correctness, completeness and to identify the quality of already developed software. Software testing is a process which is used to detect the bugs and uncover it. Software testing is a process and discipline also. It is different from software development. It should be considered that is part of software development (S., 2011). Many software application are today written as web based applications. To test the effectiveness of testing these web based applications varies widely among companies and organizations. In an era of highly responsive and interactive software processes where most of the organizations are using some form of Agile methodology, the automation of the test has been becoming a foremost requirement for the software projects. Hence test automation is to be adopted. Test automation is defined as a means to use some software tool to run repeatable tests against the application which is to be tested. For regression testing it provides that responsiveness. Software Test Automation means to automate software testing activities including the development and execution of test scripts, verification of testing requirements, and the use of automated testing tools (Andreza M. F. V. de Castro et al, 2013). Test automation supports frequent regression testing. It provides rapid feedback to developers. Test automation has virtually unlimited iterations of test case execution. Automation of tests has great support for Agile and extreme development methodologies. It has disciplined documentation of test cases. It has customized defect reporting. Defects that are missed by manual testing are found in test automation. Many automation test tools are available in the market. One such tool is Selenium. Selenium is an open source GUI based testing tool that has the capability to record and playback the test cases. Selenium can be deployed on Windows, Linux and Macintosh platforms. Selenium shows many features. It is an open source tool. Hence it is beneficial for the beginners to download and make use of it. It supports many languages like Java, Php, .net, Ruby, Python, Perl. It even supports various browsers like Internet Explorer, Google chrome, Safari, Firefox. Selenium is more flexible as compared to other testing tools as QTP.

1.1 Components of Selenium

There are three components of Selenium

1. Selenium IDE
2. Selenium RC tool
3. Selenium Grid

1.1.1 Selenium IDE

Selenium IDE provides a complete Integrated Development environment for testing. It provides recording, editing and debugging tests. It is implemented as an extension of Firefox Add-on. Selenium IDE is the...
simplest of the tools in the Selenium suite, and one of the quickest way to get started with creating scripts. But it only supports record and playback within Firefox. For the automation of other browsers, Selenium RC or Selenium 2 is to be used.

1.1.2 SeleniumGrid

Selenium Grid provides with the capability to run multiple instances of the browsers to allow running tests in parallel on multiple machines. One of the server acts as a hub, various tests contact the hub to get access to the different instances of the browser instance. It uses a hub-node concept where a single machine called a hub runs the test, but different machines called nodes will do the execution. Selenium Grid should be used in either of the following cases or in both:
1. Running the tests against different browsers, operating systems, and machines all at the same time.
2. Save time in execution of your test suites.

2. Related work

Gregory M. Kaplhammer discussed about the different types of software testing and their sub categories. The way to generate test cases, how to execute them also explained in this paper. The execution process of test cases, how to prioritize them using regressing testing and various types of related testing techniques are mentioned in this paper. How the testing is done for the graphical user interface is mentioned in this paper. On the bases of the existing techniques new techniques are proposed (Daich, 2002). Gaurav Duggal and Mrs Bharti Suri have presented the various types of regression testing techniques and their classification presented by various researchers. The explanation of selective and prioritizing test cases for regression testing is defined. Also discuss about search algorithm which is my key idea. In this paper tried to explain the complete structure of regression testing to make researcher understand its importance and scope. Regression test selection is divided into three categories and explains them. Test case prioritization problem and technique is explained. Then discussed about the search algorithm of different types their approaches and challenges (Trivedi, 2012). Jovanovic Irena discussed about the two types of software testing techniques white box and black box testing techniques. These categories are further divided into subcategories like equivalence partitioning, decision table testing, exploratory testing etc. The goal of testing and comparison between two techniques with diagrammatic representation is also mentioned in this paper. Then about the software quality assurance is also mentioned in the paper. The responsibilities of the testing are also explained in the paper. The comparison between agile versus traditional, exploratory versus scripted, manual versus software is explain in the last section of the paper. (B., 2003) Corina S. Pasareanu have introduced the new research trends in symbolic execution, how to test generation and program analysis. First describe a procedure that handles complex programming constructs such as input recursive, arrays, as well as multithreading, data structures. Also describe latest hybrid techniques to overcome some of the inherent limitations of symbolic execution, such as handling native code or availability of decision procedures for the application domain that combine concrete and symbolic execution. Finally, survey of predictive testing, invariant inference also mentioned. Some traditional application such as test generation and analysis is also discussed. Parallelizing and extending the abstraction and composition is also presented in the paper. Some new heuristics techniques that handle the traditional approaches and give beneficial results for the future are also discussed. (Păsăreanu C. S., 2009) Praveen Ranjan Srivastava has discussed about the It consists of estimating testing effort, selecting appropriate test team, designing test cases, executing the software with those test cases and examining the results produced by those executions. It indicates cost of software development is committed to testing, with the percentage for testing critical software being even higher. This paper makes an attempt using fuzzy logic to estimate reliable software testing effort. In
this paper triangular membership functions are chosen with monotonic constraints. (Srivastava P. R., 2011)

3. Selenium RC

RC stands for Remote Control. Selenium RC tool allows for the automated tests for the web applications. Selenium RC allows writing automation tests in any supported programming language. It is an open source and free tool. Selenium RC has support for almost all browsers as well as support for several programming languages like Java, C#, Ruby, Python, Perl, Javascript and PHP. Selenium RC is the sever version of Selenium. You write your tests using a programming language (say Java) and client library. Your tests issues commands which is sent to the server by the client library. The server then performs actions in the browser and reports the results back to your client. Selenium Remote Control (RC) is a server, which is written in Java, which accepts commands via HTTP for the browser. RC makes it possible to write automated tests for a web application in any programming language, which allows for better integration of Selenium in existing unit test frameworks. To make writing tests easier, Selenium project currently provides client drivers for PHP, Python, Ruby, .NET, Perl and Java. The Java driver can also be used with JavaScript (via the Rhino engine). A new instance of selenium RC server is needed to launch html test case - which means that the port should be different for each parallel run. However, for Java/PHP test case only one Selenium RC instance needs to be running continuously.

3.1 Selenium RC components

Selenium RC components are Selenium server and Client Libraries.

3.1.1 Selenium Server

Selenium server accepts commands from the test program, interprets them, executes them and sends back the result to the program. The RC server bundles Selenium Core and automatically injects it into the browser. This occurs when your test program opens the browser (using a client library API function). Selenium-Core is a JavaScript program, actually a set of JavaScript functions which interprets and executes Selene commands using the browser’s built-in JavaScript interpreter. By using HTTP GET/POST requests the Server receives the Selenium commands from your test program. This allows any language that can send HTTP request to automate the Selenium tests on the browsers.

3.1.2 Client Libraries

The client libraries provide the programming support that allows you to run Selenium commands from your own designed program. Every supporting language has its own client library. A Selenium client library provides a programming interface (API), i.e., a set of functions, which run Selenium commands from your own program. Within each interface, there is a programming function that supports each Selene command. The client library takes a Selenium command and passes it to the Selenium Server for processing a specific action or test against the application under test (AUT). The client library also receives the result of that command and passes it back to your program. Your program can receive the result and store it into a program variable and report it as a success or failure, or possibly take corrective action if it was an unexpected error. So to create a test program, you simply write a program that runs a set of Selenium commands using a client library API. And, optionally, if you already have a Selenium test script created in the Selenium-IDE, you can generate the Selenium RC code. The Selenium-IDE can translate (using its Export menu item) its Selenium commands into a client-driver’s API function calls. See the Selenium-IDE chapter for specifics on exporting RC code from Selenium-IDE.

Fig 2 illustrates the architecture of Selenium RC. It shows how the client libraries communicate with the Server by passing each Selenium command for execution. Then the server passes the Selenium command to the browser using Selenium-Core JavaScript commands. The browser, using its JavaScript interpreter, executes the Selenium command. This runs the Selenium action or verification you specified in your test script.

4. Present Work

As Selenium IDE is only compatible with the Firefox so this acts as a limitation in its usage as one can check only a small subsection of the browsers used by the users. But the
users are not limited to the use of only a single browser but make use of many different browsers as google chrome, Opera, Internet Explorer. Selenium RC tool was initially developed by Patrick Lightbody. This tool was developed basically to test these different browsers without the need to install the selenium core on the web server. Selenium RC tool acts as a proxy between the test scripts and the application under test. Selenium core instead of being installed on the server is bundled with the selenium RC tool. With the introduction of this tool developers are being provided with a chance to directly interact with the proxy by using the language of their own choice as a means to send the commands to the browsers. The main languages used to create Selenium tests are Java and C#. This is due to the fact that most web applications are being created in these languages. There is increased rate of language bindings for the dynamic languages which are created and supported mostly as the developers are moving their web applications to those languages. Languages that people are moving to are Ruby and Python.

Robustness is the main reason why preference is given to write test cases in programming languages and not using HTML-style tests with the Selenium IDE. And also to get the advantage of all the setups that are the common in most testing frameworks. Presently, Selenium RC tool makes use of various assert functions to deal with test data checking in databases. These functions are applied to different scenarios where data need to be checked in databases, such as: searching for primary keys, non-primary keys columns, or the last record inserted into a table. An interesting and very relevant aspect of this from a practical viewpoint is that it would be able to be used in real software projects. (Andreza M. F. V. de Castro, 2013)

5. Proposed Work

Selenium is one of the test automation tool that is available in the market to ensure quality of the developed software. One of the components of Selenium that is Selenium RC tool helps in testing the web applications that has direct link to the real time database entries. The proposed work is an extension of this very popular functional testing tool for user interface Selenium to support database testing in web based applications. We called this solution Selenium DB. The proposed tool implements new assert functions into the Selenium framework’s core to deal with test data checking in databases. These functions are applied to different scenarios where data need to be checked in databases, such as: searching for primary keys, non-primary keys columns, or the last record inserted into a table. An interesting and very relevant aspect of this proposed tool from a practical viewpoint is that it would be able to be used now in real software projects. In the existing work it deals with dynamic websites which are not compatible with MS Access and MYSQL and SQL Server to connect with database. But our proposed work will be able to support MS Access and Sql Server after modification in functions.

Conclusion

Many automation tools are available in the market. SIKOSA, dart, JUnit and RAGS are a few to be named. But each one has some limitations that are in some way overcome in the Selenium tool. Selenium RC tool has some assert functions that check real time user entries from the database. An extension to this tool so as to make it compatible with the some more web based applications having real time database entries is been proposed. Some new assert will be added in the Selenium RC tool so as to provide different tests in a database.

Acknowledgement

I thank my guide Ms. Neha Malhotra. Her guidance and support, and her understanding were a valuable asset during the research. I thanks for her unconditional support, for inspiring me, for her valuable insights, patience and guidance. I am very honoured to have had the opportunity to work with her. I specially thank to my parents who helped me a lot during this work.

References


Daich, G. T. (2002). Defining a software strategy. STSC.


Updated at 2014-06-13

References


Daich, G. T. (2002). Defining a software strategy. STSC.


