Automated tests on websites using Selenium-IDE.

Gustavo Antonio Toretti (gustavot@br.ibm.com), IT Specialist, IBM

Summary: Testing web sites manually is a task prone to errors. Automating tests reduces the chance to get errors and makes tests more agile. This article describes how to use Selenium-IDE in the creation and execution of automated tests for web sites.

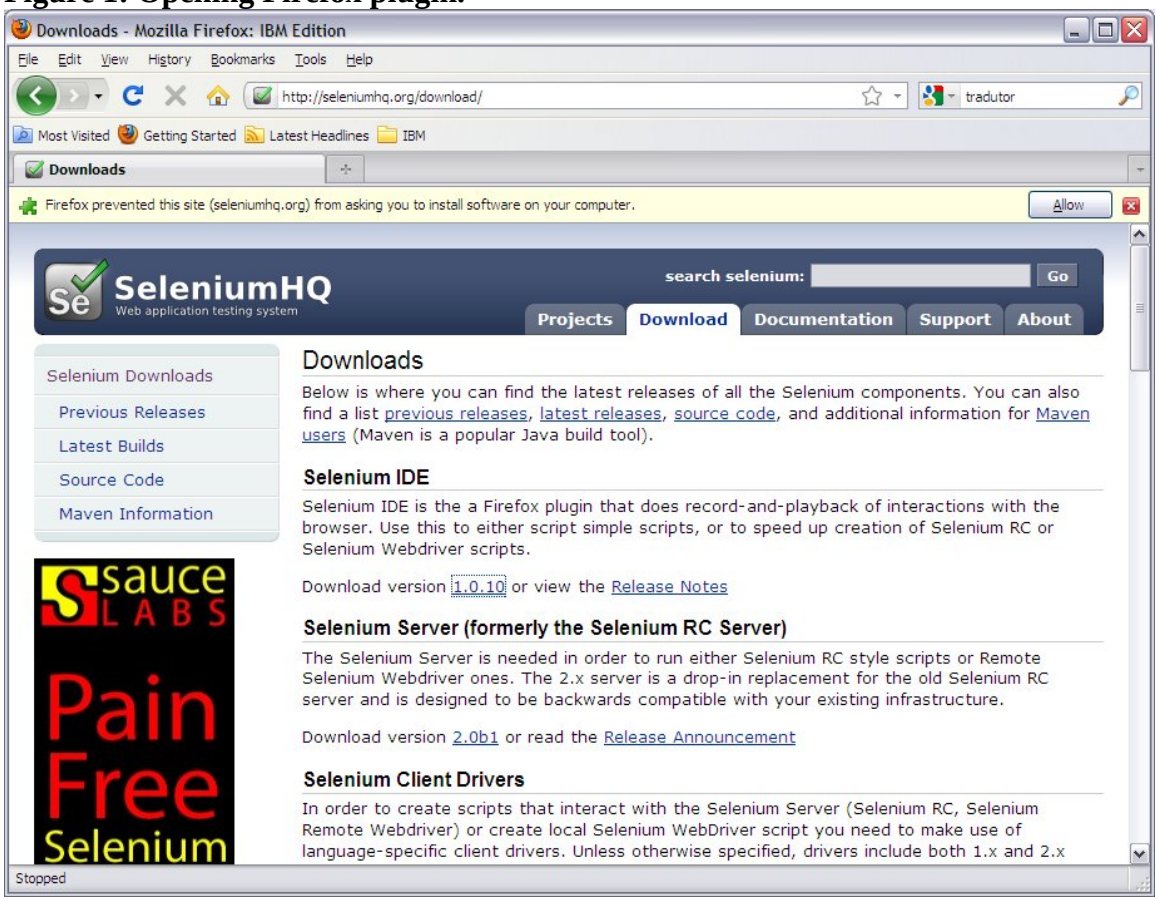
Benefits of automating integration tests

During project development, the integration testing phase involves a massive execution of system test cases, which are organized and run in a specific sequence to match system requirements.
When errors were found during execution of test cases, testing is halted until a new system version is available for testing again. The sequence of tests will be restarted on new system version and tests will continue to run until next system error appear, causing interruption of tests again, or until all tests are completed, thus ending the testing phase.
The testing phase of the project aims to ensure that the system is ready to be available to the user. Testing web systems or websites manually is a task prone to errors. Automate testing makes this task error free and does not demand tester to play the final user inputting data to the system.
This article describes a simple procedure for creation and running of automated testing of websites using Selenium-IDE, which can be used to test any system that uses the Firefox web browser. Selenium-IDE is a Firefox plugin, which can be downloaded at the Selenium website http://seleniumhq.org .

Installing Selenium-IDE on Firefox

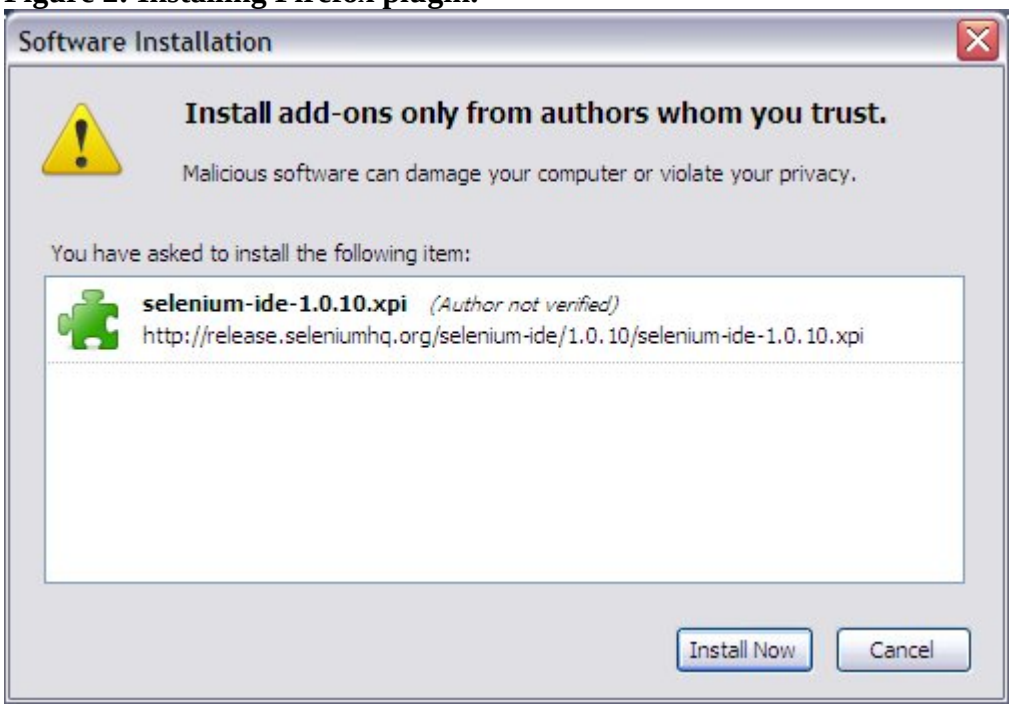
The installation file of Selenium-IDE can be obtained at downloads section of Selenium website http://seleniumhq.org/download/ in XPI file extension. Files in this extension can be opened with Firefox itself, which tries to install the plugin as described in the figure 1.

Figure 1: Opening Firefox plugin.



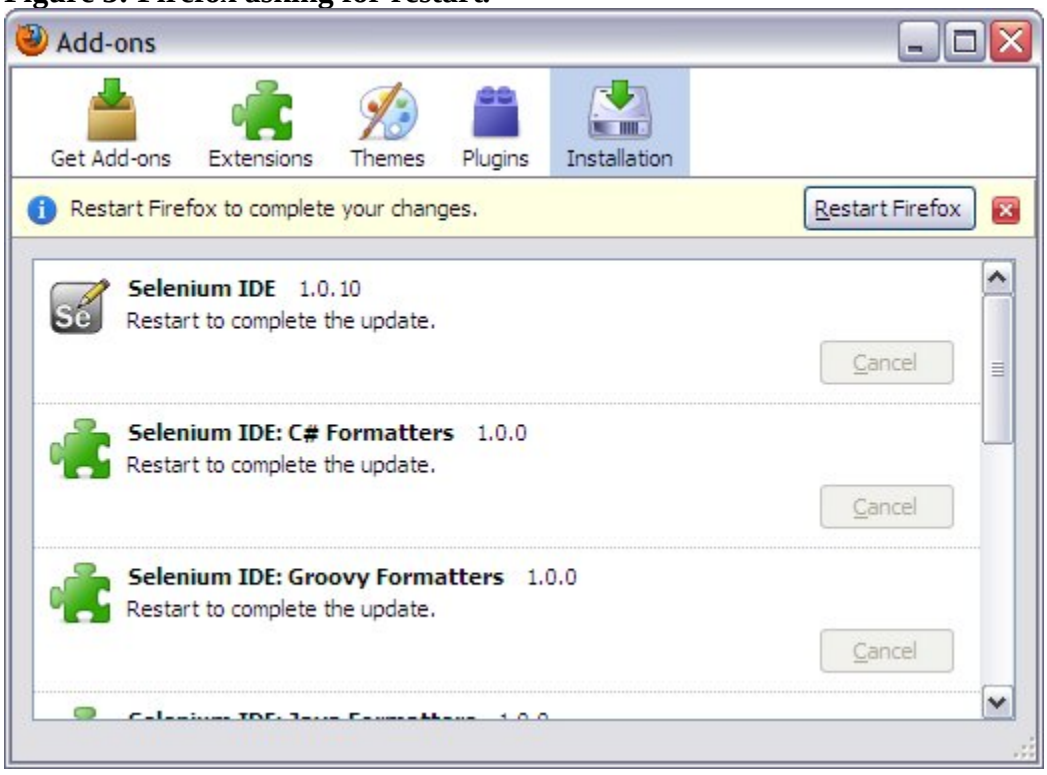
After clicking on the Allow button above, the following screen appears:

Figure 2: Installing Firefox plugin.



Next, click on the Install Now button, the plugin installation process will start. At the end, the following screen will appear prompting you to restart Firefox.

Figure 3: Firefox asking for restart.



Click at the Restart Firefox button. After restarting Firefox, Selenium-IDE can be accessed via the Tools menu, Selenium IDE.

Figure 4: Seleniun IDE menu.

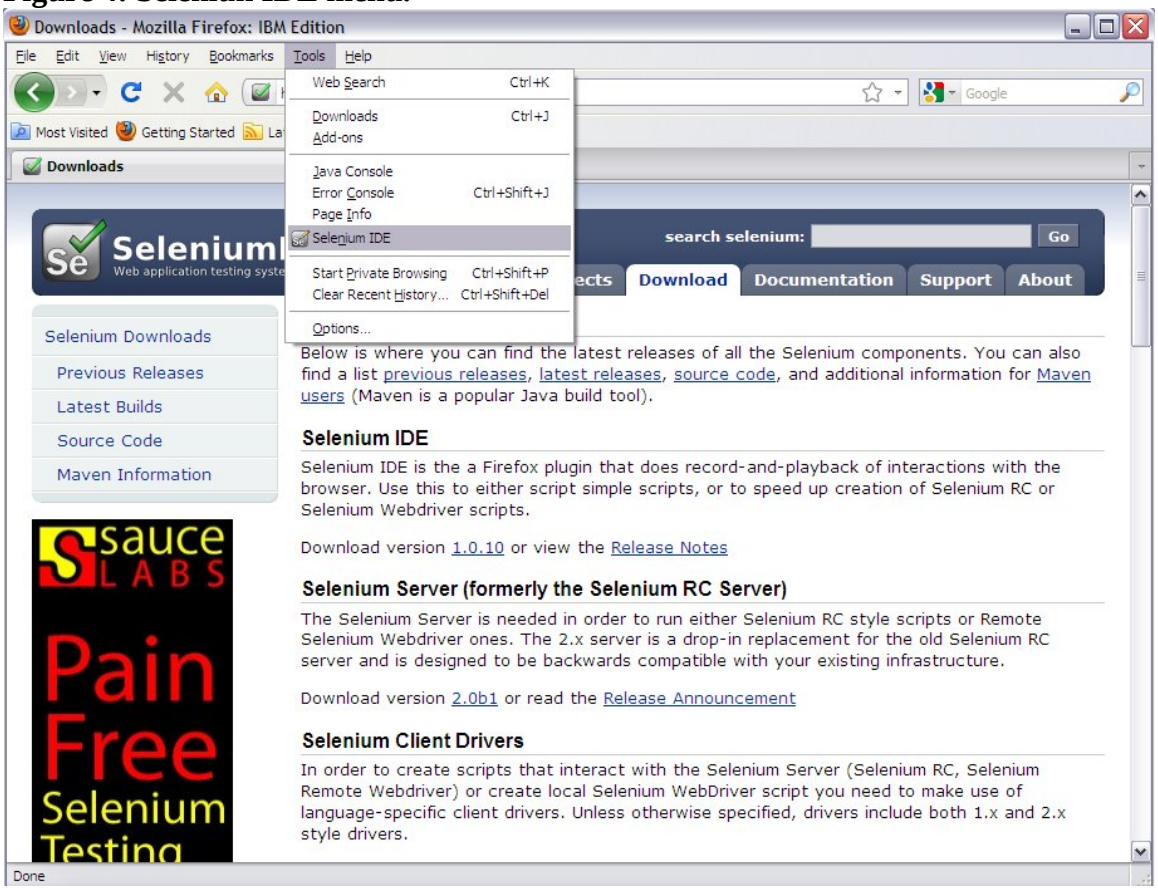
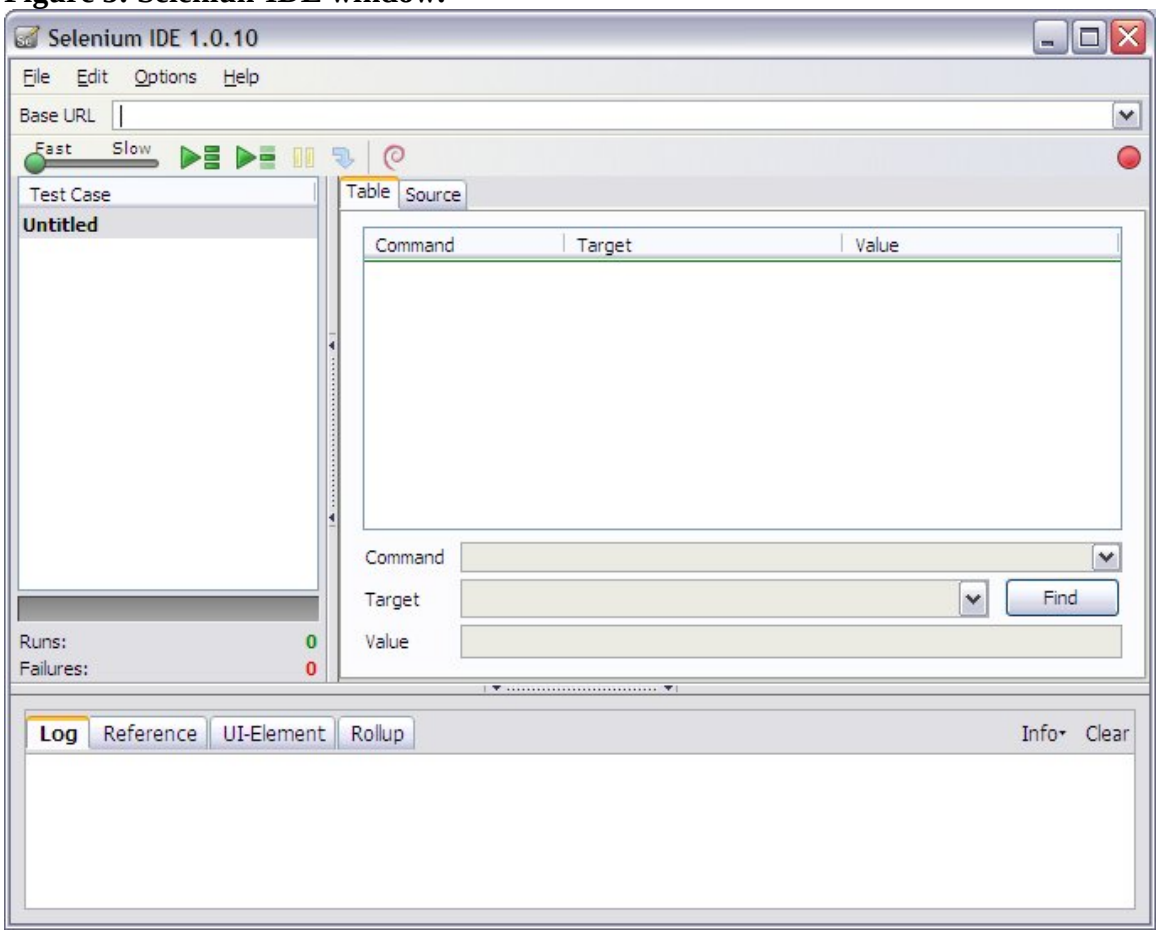


Figure 5: Seleniun-IDE window.



Creation of automated test cases

Selenium-IDE enables the creation of test cases by recording user data entries in the system page forms. It also records user actions, like clicking on buttons, links, etc. Creating automated test cases requires the user to enter the URL of the system to be tested and make the entry of data regarding the test case while Selenium-IDE records user actions.
The following example shows how to use Selenium-IDE. It will be used a sample system whose only function is to receive data and submit a page form. This sample is not available at internet, because its function is only to illustrate how test cases are created.

Figure 6: Sample of web site.

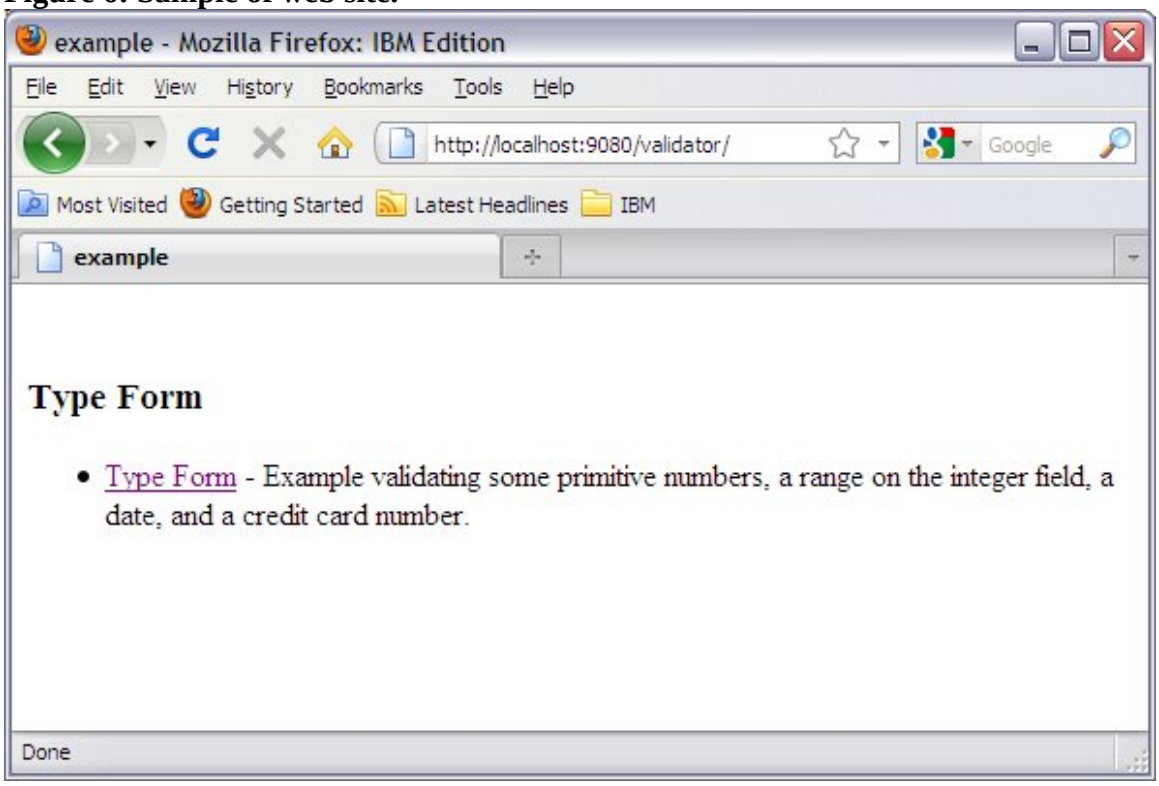
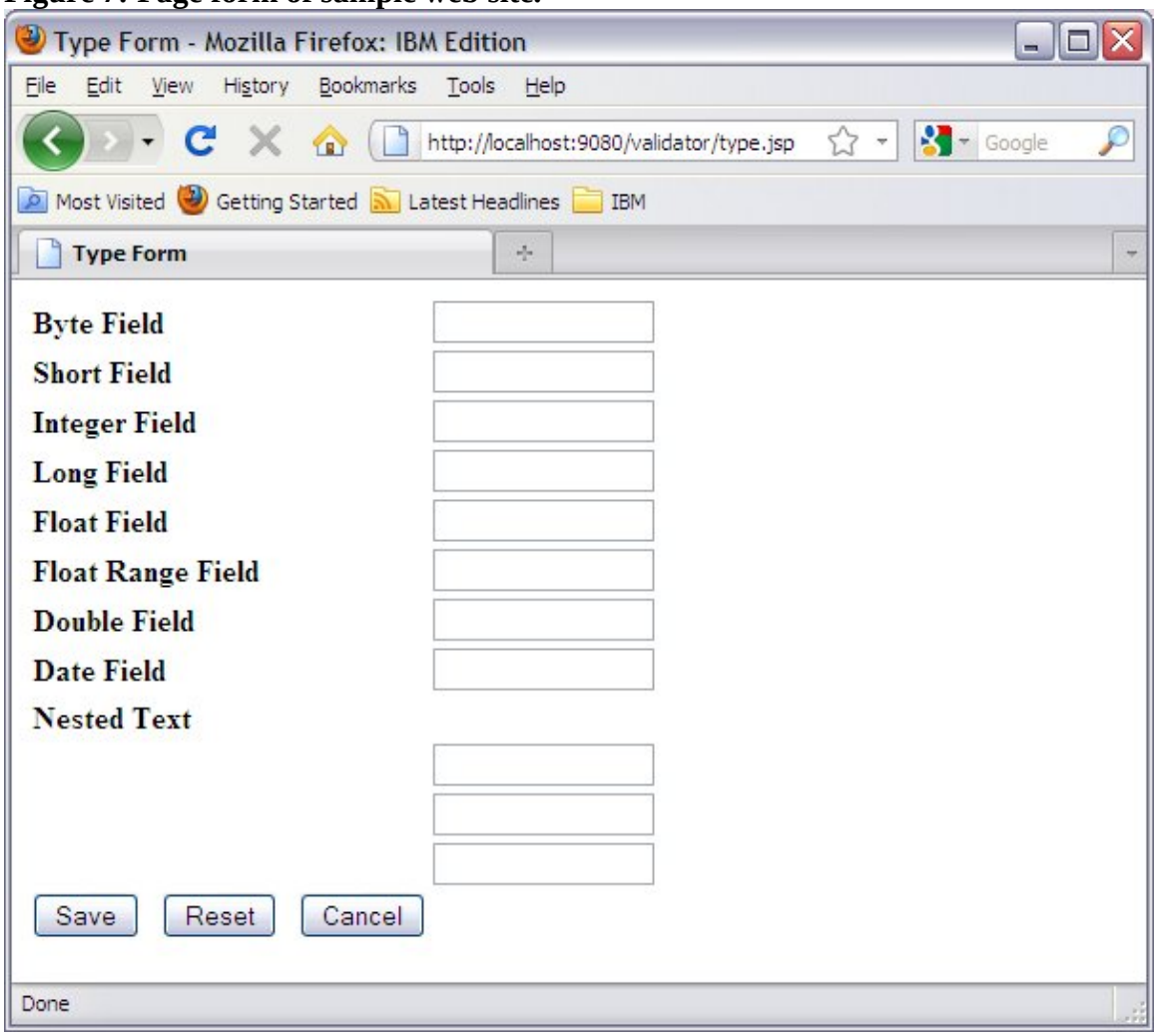


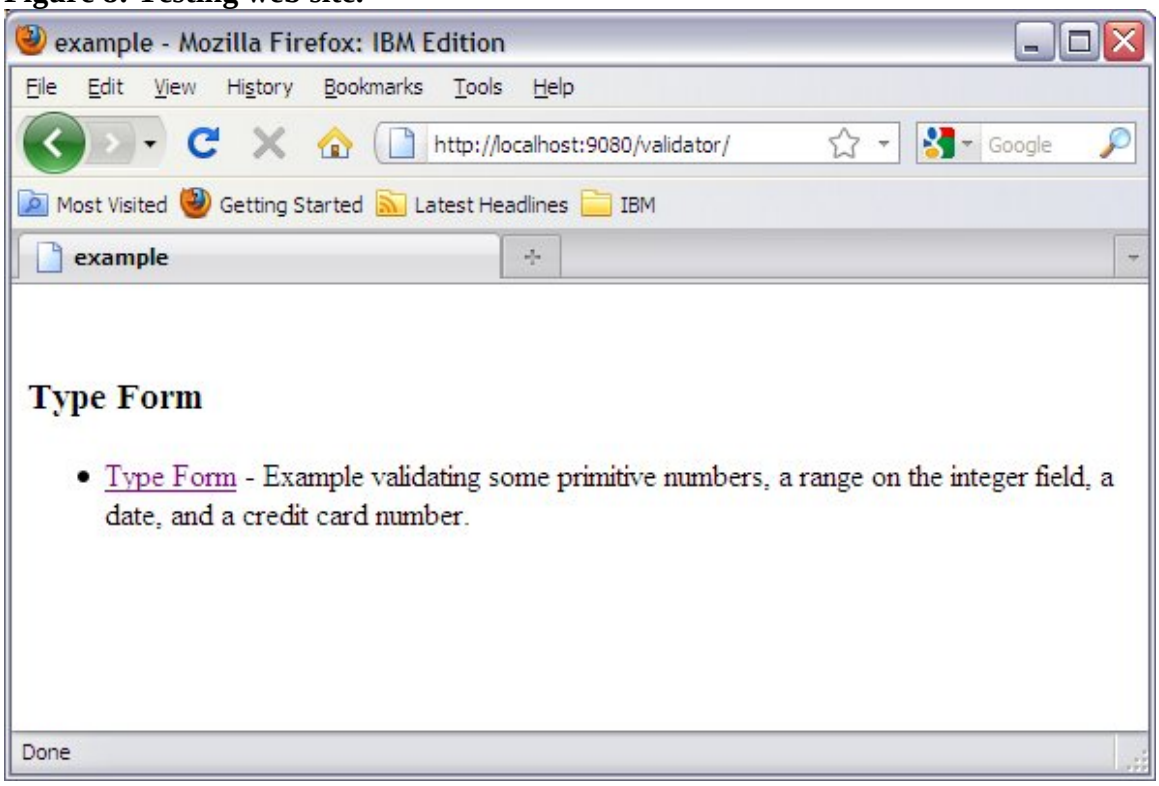
Figure 7: Page form of sample web site.



The procedure for creating automated test case consists of the following steps:

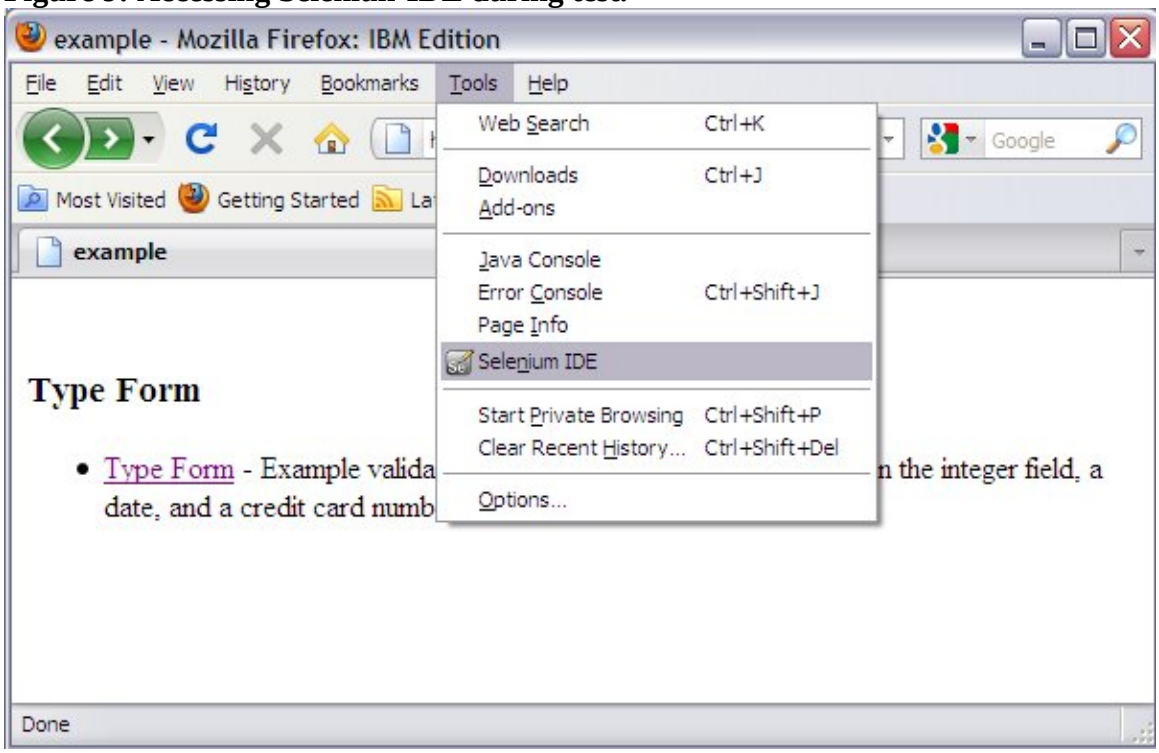
1. Start a Firefox window and navigate to the page URL you want to test.

Figure 8: Testing web site.



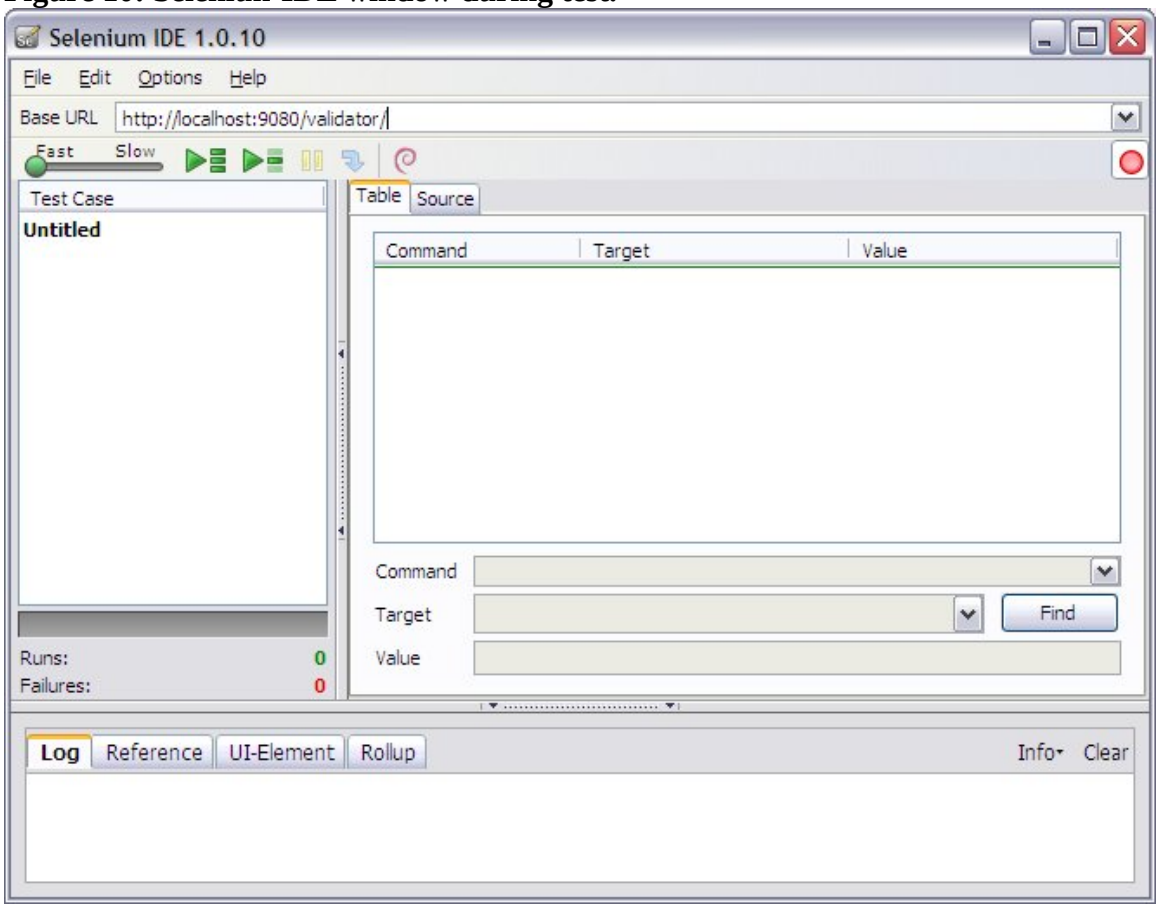
2. Click on the Tools menu, Selenium IDE, as described in the following figure.

Figure 9: Accessing Seleniun-IDE during test.



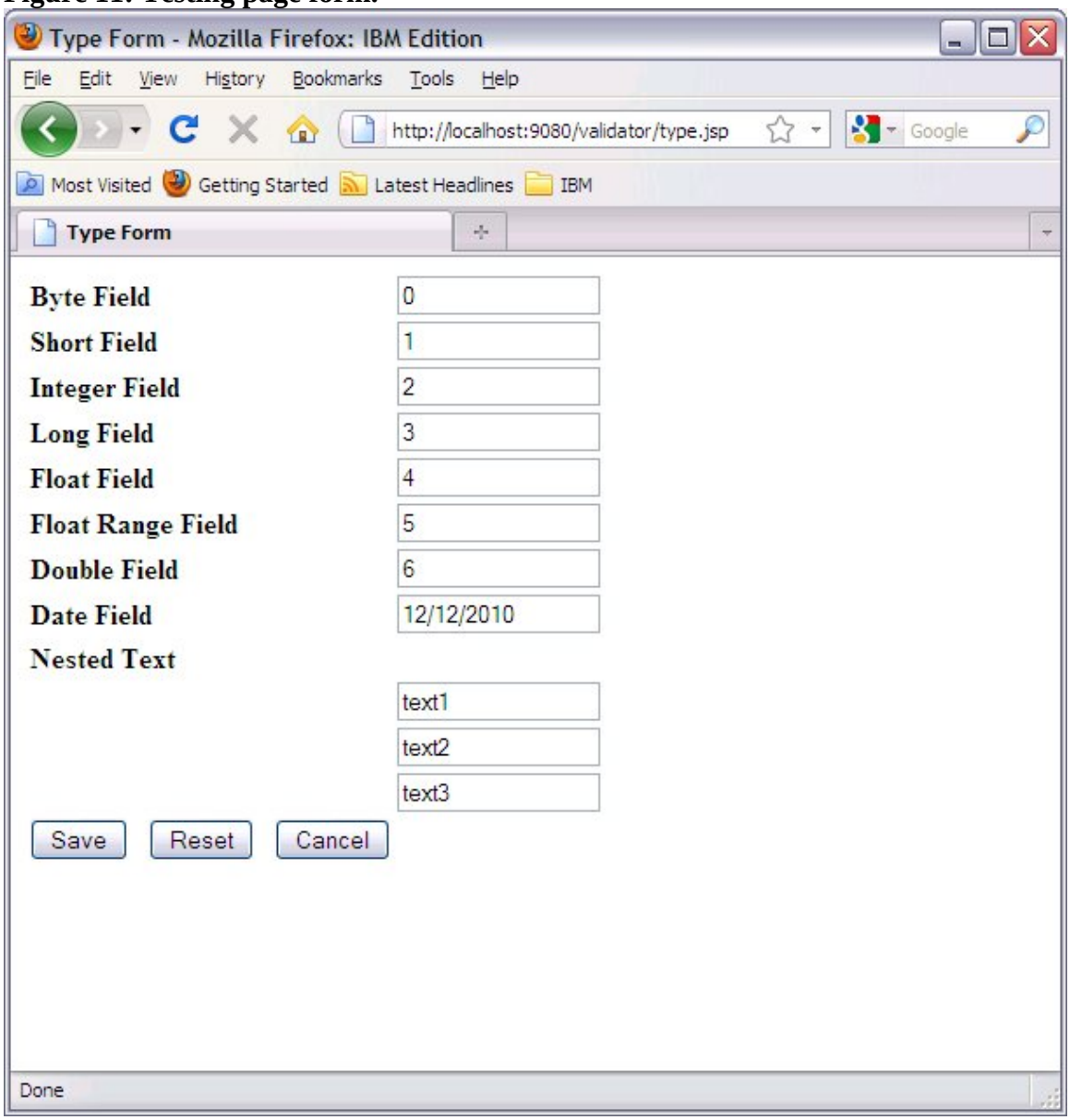
3. At the Selenium-IDE window, fill in the Base URL field at the top with the starting URL of the page being tested and then clink on the button Record  .

Figure 10: Seleniun-IDE window during test.



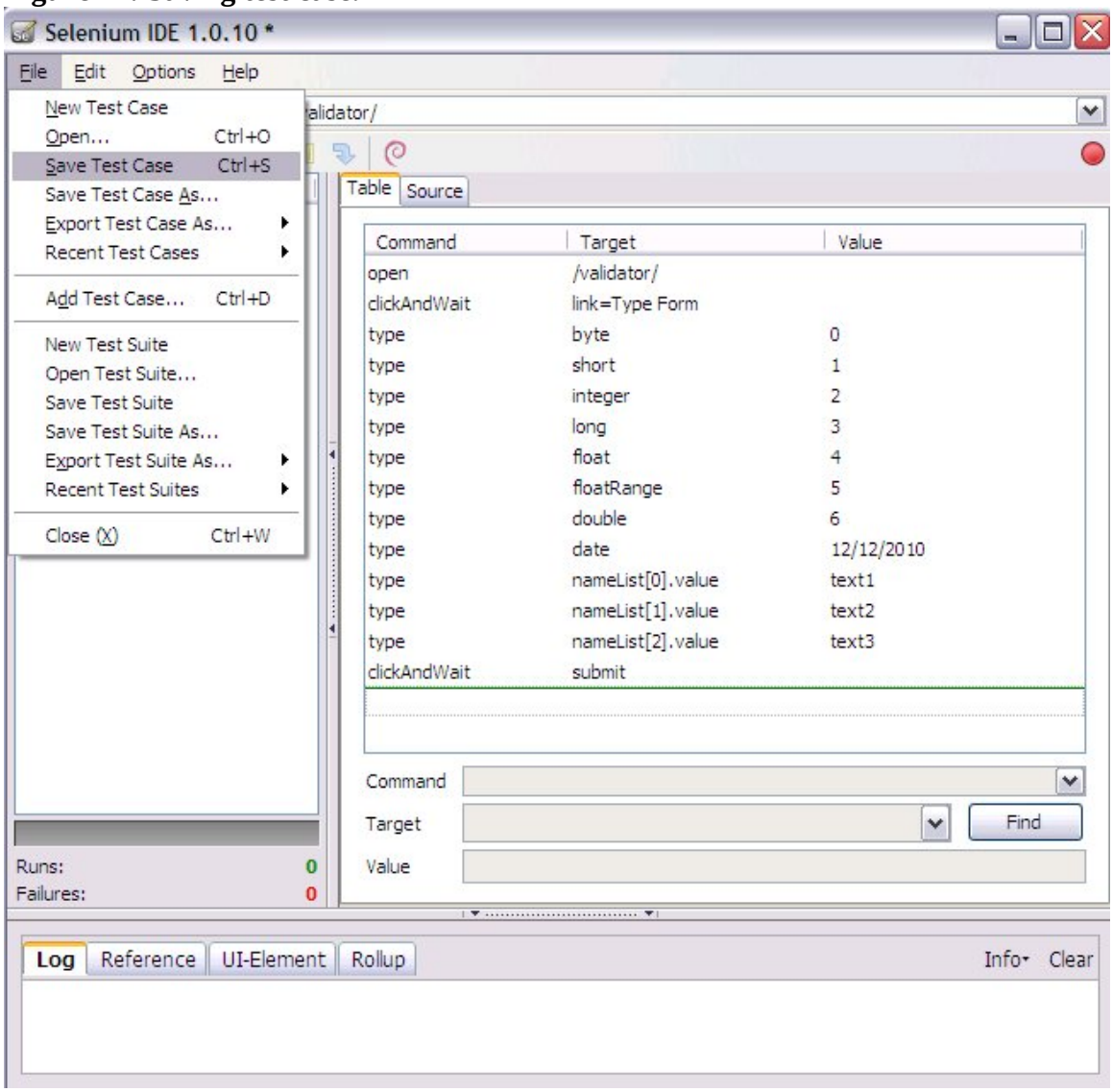
4. Having the Record button enabled, start the test case by making the data entry on the system.

Figure 11: Testing page form.



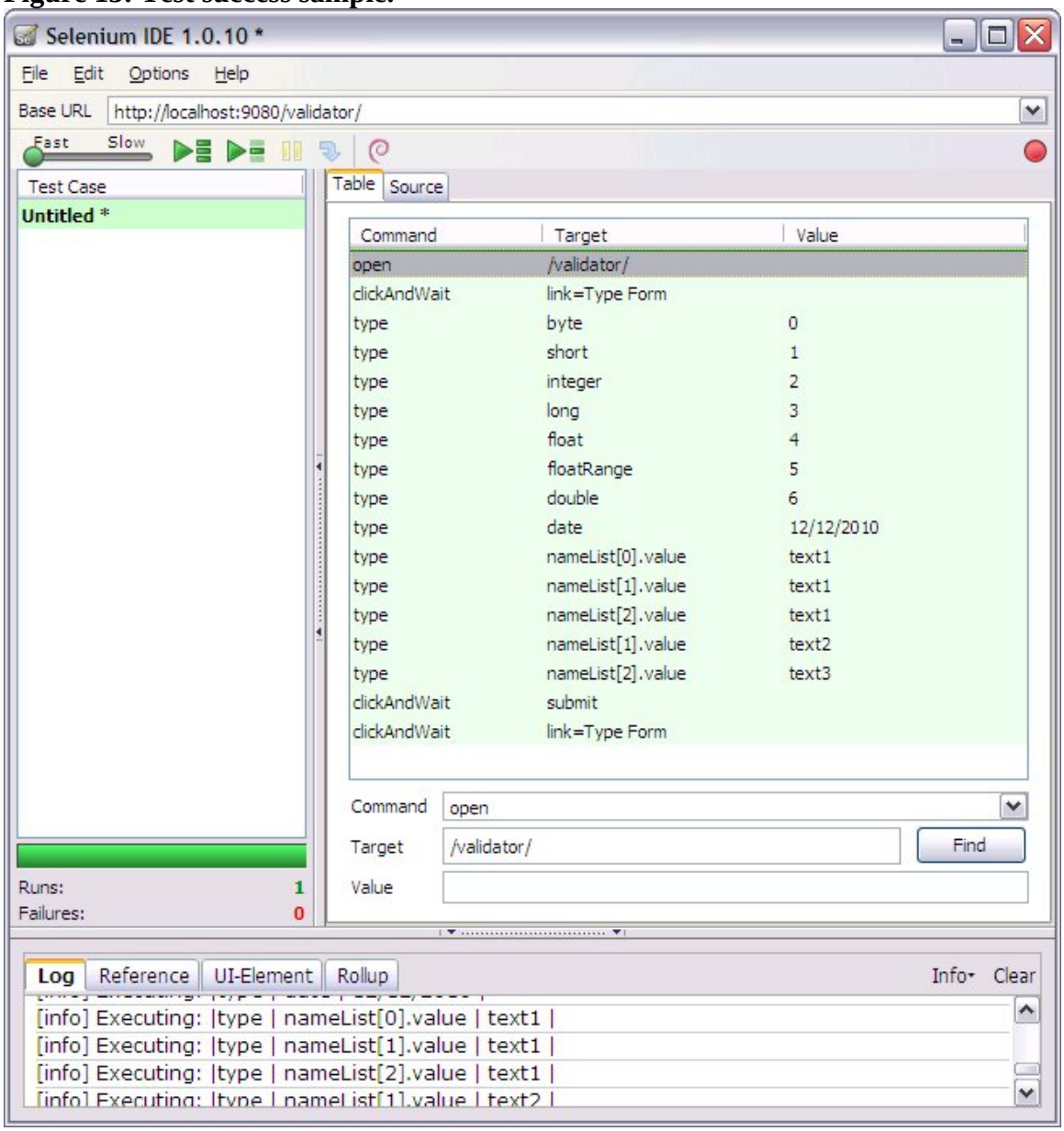
5. At the end of test case, disable the Record button. You can save the test case at the File menu, Save Test Case.

Figure 12: Saving test case.

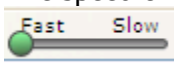


To run the test case automatically, just click on the button Play current test case  . At the end of successful execution, Selenium-IDE window will display the list of steps executed successfully.

Figure 13: Test success sample.



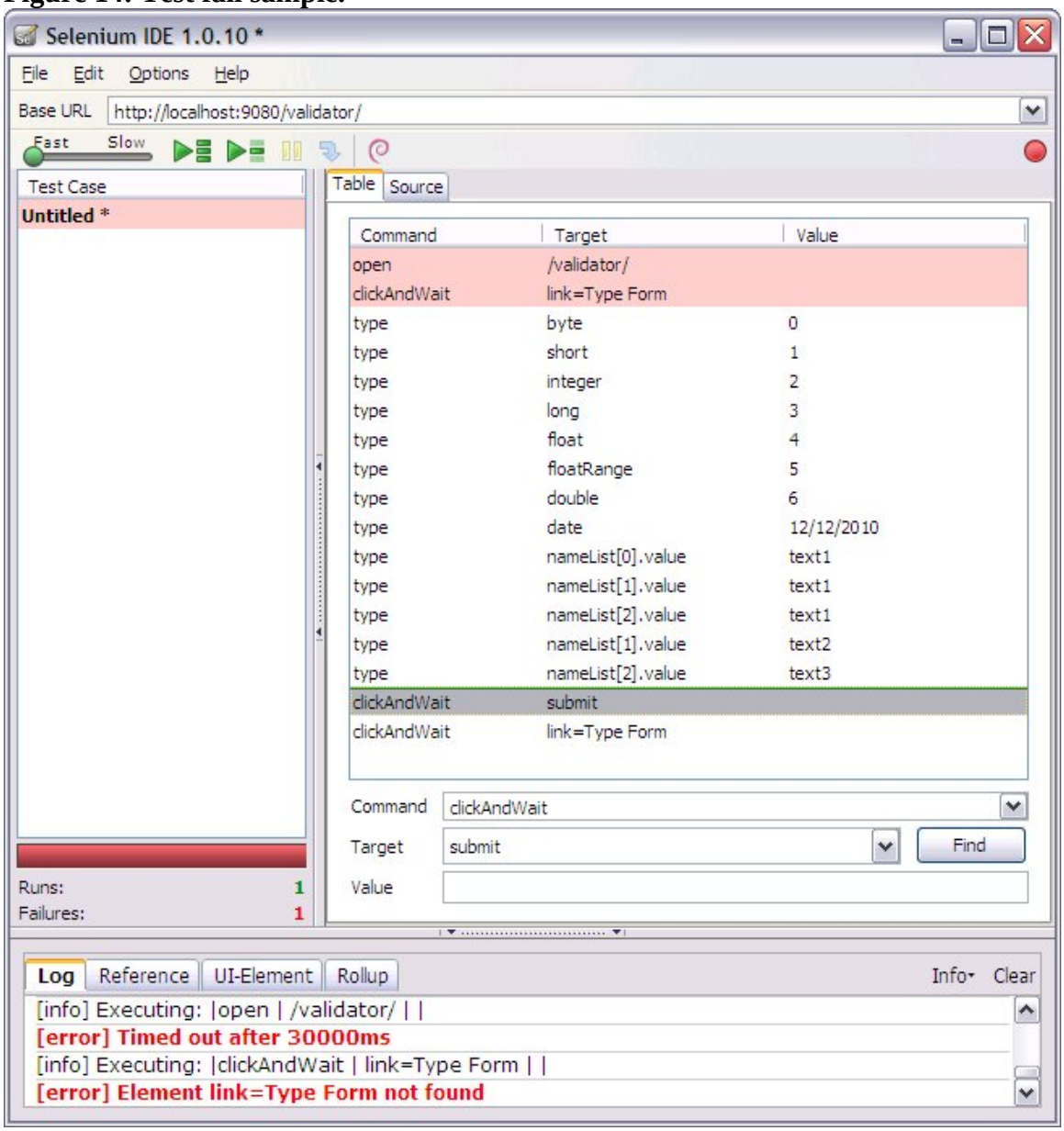
The speed of execution of test cases can be controlled through the speed control



.

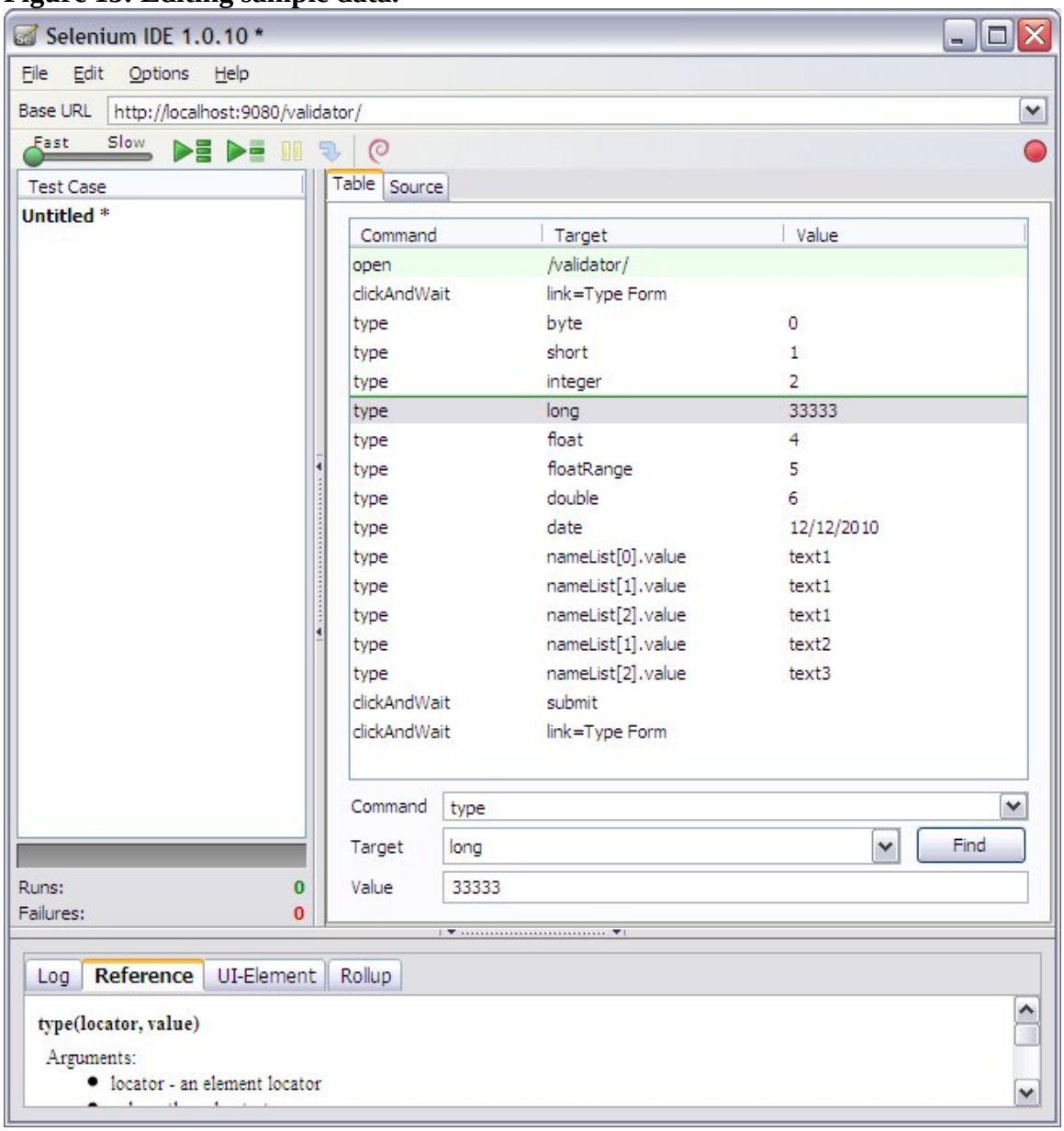
When errors occur in the execution of automated test case, the Selenium-IDE window displays the step of execution where the error occurred, selected in red.

Figure 14: Test fail sample.



It is even possible to change the data used in the automated test case. To do this simply select the pace at which certain data is inserted in the text field of the page form and then change the Value field in its properties displayed below.

Figure 15: Editing sample data.

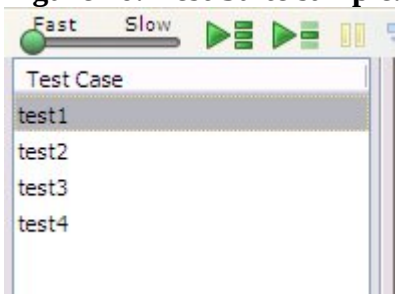


Creation of groups of test cases: Test Suite.

Selenium-IDE allows creation of groups of test cases, which are executed in the sequence they were defined inside the group. The group of test cases is defined as Test Suite.

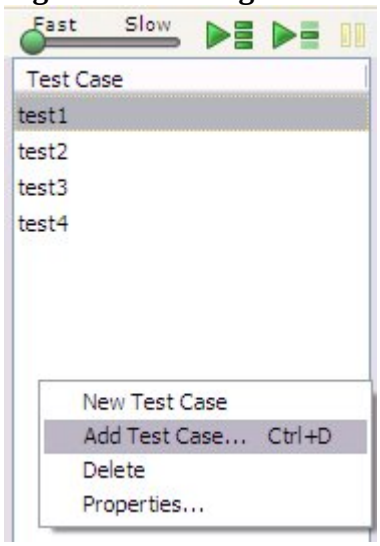
The Test Suite can be run by pressing the Play Entire Test Suite button. It will run the entire list of test cases from the Test Suite in the order it was defined in the Test Suite.

Figure 16: Test Suite sample.



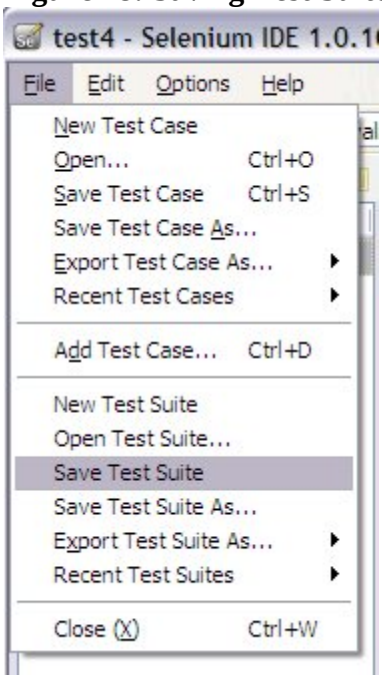
The list of test cases from Test Suite can be edited using the popup menu in the left pane.

Figure 17: Editing Test Suite.



The Test Suite can be saved using the File menu, Save Test Suite.

Figure 18: Saving Test Suite.



Once the Test Cases and Test Suites are saved, they can be run by any tester or group of testers having the Selenium-IDE available.

Conclusion

Selenium-IDE meets the need for automation of test cases for websites by using a simple solution, considering this is small software installed as Firefox plugin. Selenium-IDE provides flexibility in creating test cases allowing the organization of groups (Test Suites). Moreover, the data used in the execution of tests are editable during the execution itself, allowing the tests to be repeated with a greater variety of data, increasing tests quality.

References

- Selenium website: http://seleniumhq.org/

About the author

Gustavo Antonio Toretti (gustavot@br.ibm.com) works with Web sites and Web Application development since 1998. He is Java Certified Web Component Developer and currently works for IBM as IT Specialist, still developing Web Applications.